A VERIFICATION AND VALIDATION ASSESSMENT OF RAPID AVAILABILITY PROTOTYPING FOR TESTING OPERATIONAL READINESS

THESIS

Gurhan Tarman, 1st Lieutenant, TUAF

AFIT/GOR/ENS/01M-15

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/01M-15

A VERIFICATION AND VALIDATION ASSESSMENT OF RAPID AVAILABILITY PROTOTYPING FOR TESTING OPERATIONAL READINESS

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Gurhan Tarman, B.S.

1st Lieutenant, TUAF

March 2001

AFIT/GOR/ENS/O1M-15

A VERIFICATION AND VALIDATION ASSESSMENT OF RAPID AVAILABILITY PROTOTYPING FOR TESTING OPERATIONAL READINESS

Gurhan Tarman, B.S.
1st Lieutenant, TUAF

Approved:

Stephen P. Chambal (Chairman)	02 MAR 01 date
Jeffrey W. Lanning (Member)	02 Mar 01 date

Acknowledgements

I would like to thank my thesis advisor, Captain Stephen Chambal, for his outstanding support, assistance, and endless number of hours counseling, and reviewing my thesis effort. As a RAPTOR novice, your teaching and guidance was extremely helpful. Thank you for your professionalism.

I would also like to thank my reader, Major Jeffrey Lanning, for his inputs and thorough review of my document. Even with his demanding schedule, he found time to read my work, which resulted in increased clarity.

Finally, I would like to thank my lovely wife for all the support she gave me. I know it was very hard for you but you never gave up. I also thank you for giving me the best and most precious gift, our baby. It is the best thing happened to us while we are here. Thanks a million again.

Table of Contents

List of Figures

List of Tables

AFIT/GOR/ENS/01M-15

Abstract

Through out the world, new techniques are developed to test and evaluate systems. The Department of Defense and the US Air Force conduct numerous studies into the reliability and maintainability of current and future weapons systems in an effort to understand the reliability, maintainability, and availability (RM&A) performance of these systems. Furthermore, they must verify RM&A characteristics of systems, which are still in the development phase. The Air Force Operational Test and Evaluation Center (AFOTEC) is responsible for testing new equipment to ensure their RM&A characteristics meet specifications.

Rapid Availability Prototyping for Testing Operational Readiness (RAPTOR) is the software tool developed by HQ AFOTEC to enhance existing analysis capabilities. RAPTOR is a modeling framework allowing quick creation of RM&A models for many systems.

DoD has made verification, validation, and accreditation (VV&A) of these simulation models an official requirement. This research outlines a generic VV&A plan that uses the appropriate tools and techniques to perform model verification, validation, and accreditation. This plan is then applied to RAPTOR 5.0 and a portion of verification and validation of RAPTOR is completed and recommendations for accreditation are made.

A VERIFICATION AND VALIDATION ASSESSMENT OF RAPID AVAILABILITY PROTOTYPING FOR TESTING OPERATIONAL READINESS

1. Introduction

1.1 Overview

Through out the world, new techniques are developed to test and evaluate systems. The Department of Defense and the US Air Force conduct numerous studies into the reliability and maintainability of current and future weapons systems in an effort to understand the reliability, maintainability, and availability (RM&A) costs of these systems. Furthermore, they must verify RM&A characteristics of systems, which are still in the development phase. The Air Force Operational Test and Evaluation Center (AFOTEC) is responsible for testing new equipment to ensure their RM&A characteristics meet specifications. AFOTEC also manages a large portion of the Air Force's weapon systems operational verification, validation, and testing.

There are many circumstances that prevent AFOTEC or other users (civilian or military) to completely test equipment or weapon systems before acquisition. There are some constraints that organizations cannot ignore. Two of these constraints are time and cost. Compounding these constraints are systems, which have very long reliability requirements or are very expensive to test. It is difficult to test a system that has an expected life of decades. The equipment would often be obsolete before the test was complete. Other systems may be very costly and require destructive testing. Testing

hundreds of ballistic missiles in order to figure out their reliability is impractical because they are both too expensive and very lethal. These kinds of limiting factors are driving organizations to search for alternative ways to complete test and evaluation of new equipment. One of these alternative ways is simulation modeling which can be an easier way for predicting the system performance. However, a simulation model must go through the verification, validation, and accreditation (VV&A) process before it can be used by the Air Force. The objective of this effort is to perform a portion of verification and validation of RAPTOR, in addition to creating a generic VV&A plan that can be applied to most simulation models.

Rapid Availability Prototyping for Testing Operational Readiness (RAPTOR) is the software tool developed by HQ AFOTEC to enhance existing analysis capabilities. RAPTOR is a modeling framework allowing quick creation of RM&A models for almost any system. RAPTOR is primarily intended to be used by HQ AFOTEC analysts in support of Operational Test and Evaluation (OT&E). Since RAPTOR is so easy to use, there has been an explosion of requests from academic institutions for the tool. As stated in their web site, more than 300 commercial and government organizations also have copies of this tool [9]. RAPTOR has many advantages:

1. Allows rapid creation of reliability and availability models for any system.

2. Reduces the time required to complete a reliability and availability model from months to minutes by combining all the common elements of reliability modeling and simulation.

3. Increases modeling productivity by 99%.

4. Eliminates the need for manual calculation of reliability and availability of complex systems.

RAPTOR is a powerful and user-friendly model, which makes it very easy to quickly model a system. The graphical user interface and strong emphasis on human factors make RAPTOR a very effective reliability analysis tool.

1.2 Reliability, Maintainability, and Availability

Reliability, maintainability, and availability are concerned with a system's dependability and reparability under continued use. The definition for reliability is the probability that when operating under stated environmental conditions, the system will perform its intended function adequately for a specified time. Maintainability is defined as the probability that a failed system can be made operable in a specified interval of down time. Availability is defined as the probability that a system is operating satisfactorily at any point in time and is a measure of the ratio of the operating time of the system to the operating time plus the downtime [2]. These issues are becoming very important factors for product improvement. Companies try to impress the would-be clients by focusing on the RM&A of their products as a means of improving their competitiveness. In addition, RM&A analysis plays an integral part in the design and production of efficient, cost-effective systems.

Reliability is a very fast growing and important field in consumer and capital goods industries, in space and defense industries, and in NASA and DoD agencies. Reliability provides the theoretical and practical tools to evaluate the capability of parts, components, products, and systems to perform their required functions. They must

perform a specific function in a specified environment for a given period of time without failure.

Reliability is analyzed early in product development. Analysts use analytical methods or simulation models to represent the RM&A aspects of their systems. As systems become more complex, analytical methods cannot be used. Reliability simulation offers an alternative path for early prediction of system performance. By using reliability analysis, design alternatives can be tested before production and the best option can be selected along with removing or improving identified design problems.

1.3 VV&A Issues

Validation, verification, and accreditation (VV&A), in general terms, is a process of review, analysis, testing and approving of simulation models developed for a specific purpose. It is a methodology, which helps ensure the production of quality simulation results.

VV&A is commonly used as a single expression; this is not to infer that the methodology is an all or nothing process. The objective of VV&A is to ensure the correctness, and consistency of the final product. VV&A may be performed by an independent agent, a VV&A-team, by the person(s) producing the product, or a combination of all. Model VV&A focuses on the prevention and the detection of errors, i.e. deviation from intent. This is accomplished using both manual and automated techniques. Errors include deficiencies such as unsatisfied requirements, the inclusion of extraneous functions, or using incorrect solution algorithms. An error may be in the coding of the model, the specification of the model, or the documentation. The error

might be related to the functional correctness or another property, such as performance, or a more subjective attribute such as product quality.

The motivation of testing a simulation model is to affirm model accuracy with methods that can be economically and effectively applied to both large scale and small-scale systems. One validation and verification approach does not fit all sizes of simulation projects. Different size simulation projects require different structures, different personnel, and different methodologies. A small project may only require a single analyst to obtain usable results, while a large project with various quality attributes may require several groups of analysts with different skills.

1.4 Problem Statement

Reliability simulation offers an easier path for evaluating RM&A characteristics of increasingly complex systems. The decisions being made with the simulation model are of particular importance. One of the most difficult problems facing a simulation analyst is that of trying to determine whether a simulation model is an accurate representation of the actual system being studied, i.e. whether the model is valid. In addition, the simulation model must be accredited by the using organization. Therefore, verification, validation, and accreditation process is critical to product evaluation and the final evaluation of the system.

As these systems advance so must the reliability, maintainability, and availability analysis capability. The industrial community and the Air Force continue to use simulation methods for handling RM&A analysis. AFOTEC depends on RAPTOR to evaluate new Air Force systems. They are also depending on the upgrade of RAPTOR as

a valid simulation program. The Air Force currently uses RAPTOR 2.0 to provide RM&A analysis on new equipment acquisitions. This model was verified, validated, accredited on 30 January 1996, however its capabilities are limited, and new systems require enhanced capabilities.

RAPTOR 5.0 is an enhanced simulation model, which is intended to replace RAPTOR 2.0. This change, however, requires RAPTOR 5.0 to be verified, validated, and accredited in accordance with DoD and Air Force Instructions. A generic VV&A plan must be developed and applied to RAPTOR. This research will:

1. Explore VV&A in the literature to gain more knowledge about the VV&A techniques and policies that are currently used.
2. AFI 16 – 1001 identifies the verification and validation agent's responsibility to develop a VV&A plan [10]. A generic plan will be developed that can be used for future VV&A purposes. This plan will be based on the Air Force Instruction and the insight provided by the literature review.
3. Perform a partial verification and validation of RAPTOR simulation model by applying appropriate portions of this VV&A plan and provide recommendations for accreditation.

1.5 Thesis Outline

The objective of this thesis effort is to outline a generic VV&A plan that uses the appropriate tools and techniques to perform model verification, validation, and accreditation. This plan will be applied to RAPTOR 5.0 and a portion of verification and validation of RAPTOR will be completed and recommendations for accreditation will be

made. This thesis is organized into chapters according to subject areas. Chapter 2 presents a background on VV&A issues, an overview of RM&A, and an introduction to RAPTOR. Chapter 3 presents a generic VV&A plan, tools and techniques for verification, validation, and accreditation of simulation models. Chapter 4 will apply the VV&A structure provided in Chapter 3 to the RAPTOR model. The partial RAPTOR verification and validation will be completed and documented according to the test plan. Recommendations for accreditation will be given but an overall and complete VV&A is outside the scope of this effort. Chapter 5 represents the conclusions of this thesis, recommendations for improving RAPTOR and areas for future study.

2. Literature Review

2.1 Reliability, Maintainability, and Availability Overview

Reliability, maintainability, and availability (RM&A) analysis form the foundation of the reliability engineering field. New and advanced studies continue to be an important area of research. Improved methods like reliability simulation provide analysts a technique for handling system evaluations easily in the area of reliability.

The evolution of reliability started with the mechanical systems. As the electrical systems started to be widely used after the first quarter of the twentieth century, the importance of reliability started to increase. At the early stages the reliability of the electrical systems was improved by simply adding redundant components and wiring. World War II, Korea, and the Vietnam War changed the military focus to weapon systems reliability. The explosion in electronics after the 1950s made reliability a valued branch of engineering. With the use of new and widely applied techniques in reliability, it became a part of the total quality initiatives of companies trying to gain a competitive edge.

Kececioglu states that reliability has the following objectives [2]:

1. Determine if the performance of components, equipment, and systems is within specifications for the desired function period. If it is not, find out whether it is the result of a malfunction or of a failure, which requires corrective action.

2. Determine the pattern of the failures, causes, and time to failure distributions.

3. Determine the failure rate, the mean life, the reliability of components, equipment, and systems, and their associated confidence limits at some desired confidence levels.

4. Based on the results obtained, provide guidelines for corrective actions that should be taken.

5. Reevaluate the performance of the units after corrective actions taken to assure that these actions were the correct ones and as effective as intended.

6. Determine the growth in the mean life and the reliability of units during their research, engineering, and development phase.

7. Provide a means to statistically and scientifically determine if a redesign has indeed improved the failure rate, mean life or reliability of components or equipment with the desired confidence.

8. Provide a statistical and scientific means to determine which one of the two manufacturers of a component or equipment or design should be preferred from the failure rate, mean life or reliability point of view while all other factors being practically and economically the same.

9. Select the best reliability test from the point of view of the test time, risk levels, number of equipment, cost, and personnel available to demonstrate the specified or desired failure rate, mean life or reliability at the chosen confidence level.

10. Provide management with the reliability test results, as requested by them, and in the format that will convey the requested information in a very easily understood form so they can make the right decision.

If systems are simple or not very complicated in size, they can be evaluated by using analytical methods. However, these methods are limited and cumbersome if the systems become more complex and larger in size.

2.2 Analysis Techniques

2.2.1 Analytical Solutions

Systems are generally broken down into sub-systems of components for RM&A analysis. There are several categories of component systems. The common ones include series, parallel, series-parallel, and complex systems. The simple example of a series system contains two components as shown in Figure 1.

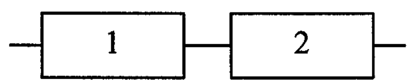

Figure 1. Series System

The reliability of a component (p) ranges from 0 to 1. Given the reliabilities of the two components are p_1 and p_2 respectively and the assumption of independently operating components, then the system reliability function $h(p)$ for the series system is:

$$h(p) = p_1 * p_2 \qquad (2.1)$$

A two component parallel system is shown in Figure 2.

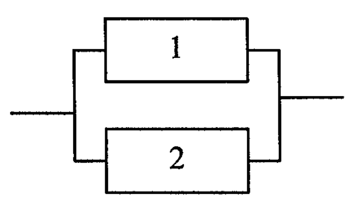

Figure 2. Parallel System

In a parallel system the reliability function is:

$$h(p) = 1 - [(1 - p_1) * (1 - p_2)] \qquad (2.2)$$

Series-parallel systems consist of combinations of series and parallel components in the system.

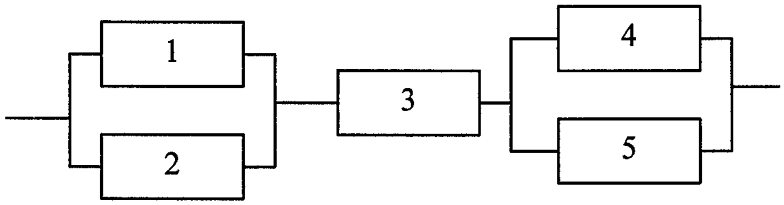

Figure 3. Series-Parallel System

The reliability function for this system is:

$$h(p) = [1 - (1 - p_1) * (1 - p_2)] * p_3 * [1 - (1 - p_4) * (1 - p_5)] \qquad (2.3)$$

An example for complex structure can be shown with the structure in Figure 4.

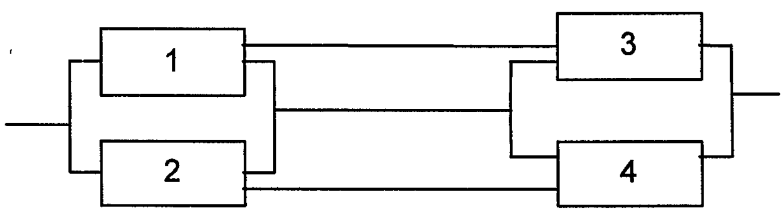

Figure 4. Complex Structure

The system reliability function for the complex structure is:

$$h(p) = ((p_1 * p_3) + (p_1 * p_4) + (p_2 * p_3) + (p_2 * p_4)$$

$$- (p_1 * p_2 * p_3) - (p_2 * p_3 * p_4) - (p_1 * p_3 * p_4) - (p_1 * p_2 * p_4) + (p_1 * p_2 * p_3 * p_4)) \qquad (2.4)$$

As it can be seen from above, the complexity of the reliability function increases as the size of the system increases. There are several analytical methods for determining the steady-state properties of systems, including Network Theory, Markov Models, Cut

Sets, Venn Decomposition, and Non-Homogenous Poisson Process. Analytical methods can directly solve many reliability systems and provide exact solutions. However, if the complexity of the system increases, analytical solutions become cumbersome and impractical. In these kinds of situations where analytical methods are inadequate, simulation provides a dependable and preferable alternative. The results of simulation models approximate system performance. Although analytical methods provide exact solutions, a reasonable approximation through simulation is better than no solution when analytical methods fail. As Law and Kelton state, the accuracy of the results is getting better [13]. Although the accuracy is getting better, there is still need to improve the modeling capability of current simulation tools to account for realistic behavior of real-world systems.

2.2.2 Simulation Analysis

Simulation analysis is an alternative to analytical models and solutions. Most of these simulation tools are based on Monte Carlo simulation. Monte Carlo simulation makes consecutive draws from varying distributions representing the failure and repair rates. By using these distributions, the simulation predicts the actual system performance. Monte Carlo simulation offers a reasonable solution for determining reliability, maintainability, and availability to complex systems when analytical methods are inadequate. This technique applies to almost all types of systems, from simple to complex reliability block diagram. Reliability block diagram simulation has a great potential for predicting large-scale system reliability and availability.

Simulation analysis can be used in two different ways. In the first choice, the users create their own simulation model by writing their own code. This can be a tough choice if one does not have enough experience with a high-end simulation language. Even if the users are capable of code writing, they still have to be sure that their model is working correctly and they have built the right model for their needs.

The second choice is an easier approach. An analyst can use a verified and validated simulation model. In this case, they do not have to worry about the VV&A issues. These software products are much more user-friendly and give more flexibility and speed to the user. Some examples of reliability simulation models are Relex, Item, Avesim and RAPTOR.

2.3 Verification, Validation, and Accreditation

Military weapon or defense systems are often large and extremely complex. They are difficult to understand and evaluate without the help of modeling and simulation. Discrete event and time-stepped simulations have been the foundations for the major computer tools for analyzing combat since the 1960s. Decisions and actions involving billions of dollars have been and will continue to be influenced by these simulations. Realizing this, DoD has made verification, validation, and accreditation of these simulations an official requirement [7].

According to Air Force Instruction 16-1001, verification is the process of determining that modeling and simulation (M&S) accurately represents the developer's conceptual description and specifications. This is accomplished by identifying and eliminating mistakes in logic, mathematics, or programming. This process establishes

that the M&S code and logic correctly perform the intended functions. AFI 16-1001 defines validation as the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model [10]. Validation process can also be used to identify necessary model improvements. Validation has two major components. The first part is structural validation. This includes an internal examination of M&S assumptions, architecture, and algorithms in the context. The second part is output validation, which determines how well, the M&S results compare with the perceived real world. Accreditation is the official determination by the accreditation authority whether or not the modeling and simulation is acceptable for a specific purpose. This determination considers the verification and validation status of a specific model version, its data support, and the analyst or the users that operate the model. The accreditation authority is the individual who is responsible and accountable for decisions or actions based upon the specific modeling and simulation usage. The decision to accredit a model or simulation rests only with the accreditation authority. The verification and validation documentation of model will be reviewed in the later sections of this effort.

According to AFI 16 – 1001 the key actions and elements in Figure 5 constitute a flexible process that accommodates VV&A activities throughout the M&S' life cycle. Even though the Air Force and other experts in this area have their own definitions and techniques for VV&A, there is no general VV&A plan applicable to every simulation model.

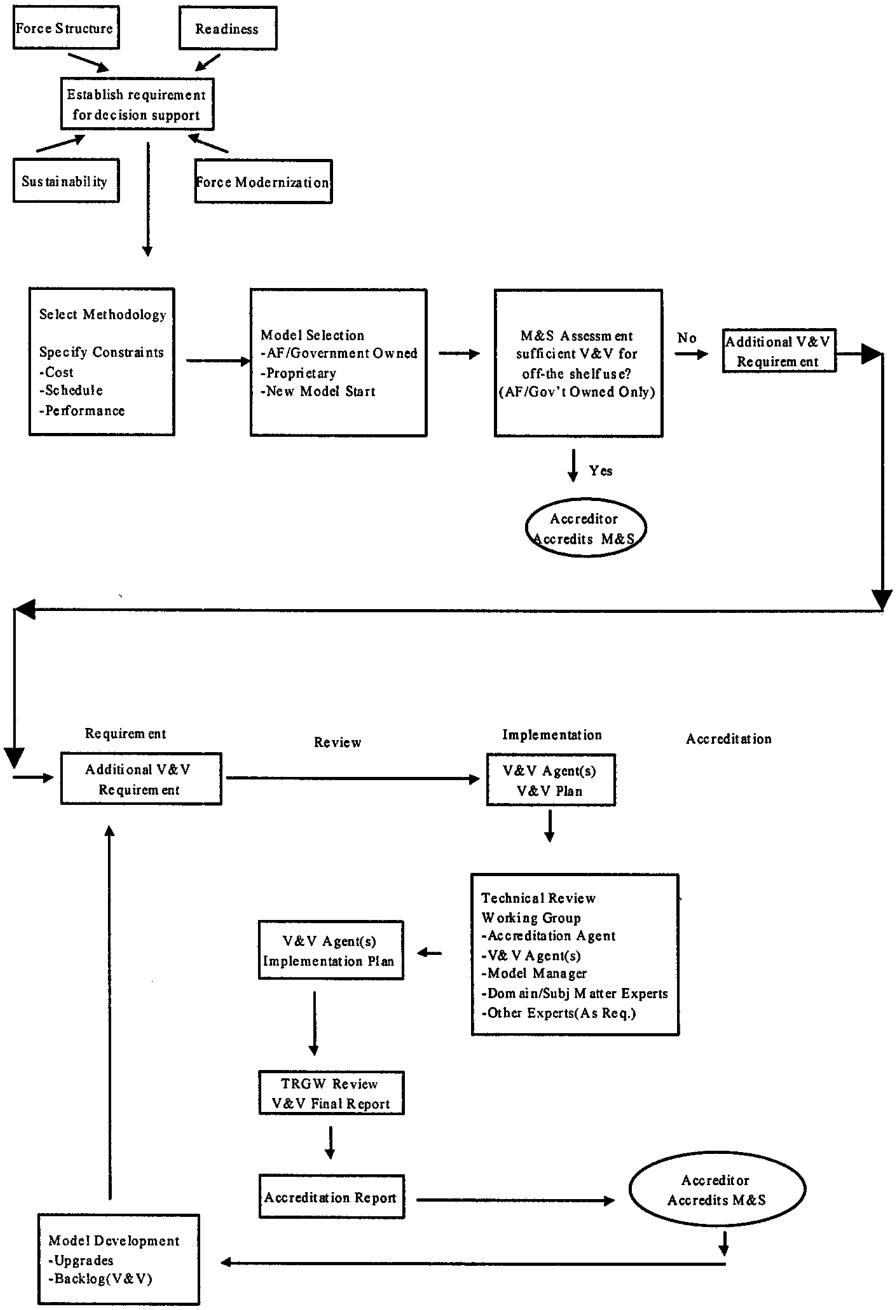

Figure 5. Air Force VV&A Process

In this effort, the objective is to come up with a new VV&A plan, which can be applied to general modeling and simulation that can be used by the Air Force. The

foundation of this plan will be situated over the Air Force VV&A process, but it would also be applicable to models used outside the Air Force.

The first part of Figure 5 shows the steps accomplished during the development phase of the model. The second part introduces the VV&A effort accomplished by the agent using the VV&A plan. The agent should follow some rules and principles during this process.

Verification and validation are based on certain underlying principles. According to the Webster's dictionary, a principle is defined as:

> 1. An accepted or professed rule of action or conduct. 2. A fundamental, primary or general laws or truth from which others is derived. 3. A fundamental doctrine or tenet; a distinctive ruling opinion.

Understanding and applying these principles is very important for the success of a modeling and simulation effort. Principles help the researchers, agents and managers to better understand what validation and verification is about.

According to Balci, et al, the following principles can be helpful during the VV&A phase of a modeling and simulation [14]:

1. Verification and validation must be conducted throughout the entire modeling and simulation life cycle.

2. The outcome of verification and validation should not be considered as a binary variable, meaning the model or the simulation is absolutely correct or absolutely incorrect.

3. A simulation model is built with respect to the modeling and simulation objectives and its credibility should be judged with respect to those objectives.

4. Verification and validation requires independence and objectivity to prevent developer's bias and influence.

5. Verification and validation is difficult. It is more difficult if verification and validation does not start with the development of the model.

6. Complete simulation model testing is not possible.

7. Verification and validation must be planned and documented.

8. Errors must be prevented. (Type I, Type II, Type III)

9. Errors should be detected as early as possible in the modeling and simulation life cycle. This prevents costs of revising the model in the later stages of development.

10. Multiple response problems must be recognized and resolved properly.

11. Successful testing of each module does not imply overall model credibility. The interaction between the modules must also be tested.

12. Simulation model validity does not guarantee the credibility and acceptability of simulation results.

13. A well-formulated problem is essential to the acceptability and accreditation of modeling and simulation results.

Verification and validation is not a phase or step in the modeling and simulation life cycle, but a continuous activity throughout the entire life cycle as announced in principle 1. Conducting VV&A for the first time at the end of the M&S application life cycle would be like a teacher who only gives a final examination. The student does not know if she or he has any deficiencies until the course is complete. Frequent tests and

homework through the quarter notifies the student about his or her deficiencies so they can study more and improve their knowledge about the course.

The situation in conducting verification and validation is similar to this example. VV&A activities throughout the entire M&S life cycle are intended to highlight any quality deficiencies. This helps the researchers early on to improve the product during the life cycle. Errors are detected as early as possible. Delaying verification and validation to later stages increases the possibility of making Type I and Type II errors. Type I error is done when the model is rejected even though the model is valid. Type II error is made when the model is accepted even though the model is not valid.

2.4 Verification, Validation, and Accreditation Process

Three basic approaches are used in deciding if a simulation model is credible or not. Each of the approaches requires the model development team to conduct verification and validation as a part of the model development process. The most common approach is to have the development team make the accreditation decision. This is a subjective decision based on the results of the various tests and evaluations conducted as a part of the model development process.

Another approach is independent verification and validation (IV&V), which uses a third party to decide whether the model is credible. The third party is independent of both the model development team and the model sponsor. After the model is developed, the third party conducts an evaluation to determine its credibility. Based upon this evaluation, the third party makes a subjective decision on the accreditation of the model.

This approach is usually used when a large cost is associated with the problem requiring the simulation model.

The IV&V approach ranges from simply reviewing the verification and validation conducted by the model development team to a complete verification and validation effort. A complete IV&V evaluation is extremely costly and time consuming. If a decision is made to use IV&V, it should be early in the model development process. This decision would significantly decrease the cost of VV&A. If the model has already been developed, the IV&V agent should only evaluate the verification and validation that has already been performed before. Starting the VV&A process after the model has been developed would be very time consuming and this again increases the cost.

The last approach to determine model validity is to use a scoring technique. Scores or weights are determined subjectively during the validation process. Then they are combined to determine the category scores and the overall score for the simulation model. A simulation model is considered valid if its overall and category scores are greater than a certain predetermined score.

Sargent thinks that this approach is impractical due to the subjective passing scores. In addition, the model may receive a passing score but it may still have a deficiency that needs to be corrected [15]. These are the three main approaches that are used. We should also understand how model verification and validation relate to the model development process. A simple model development process can be used as an example like in Figure 6.

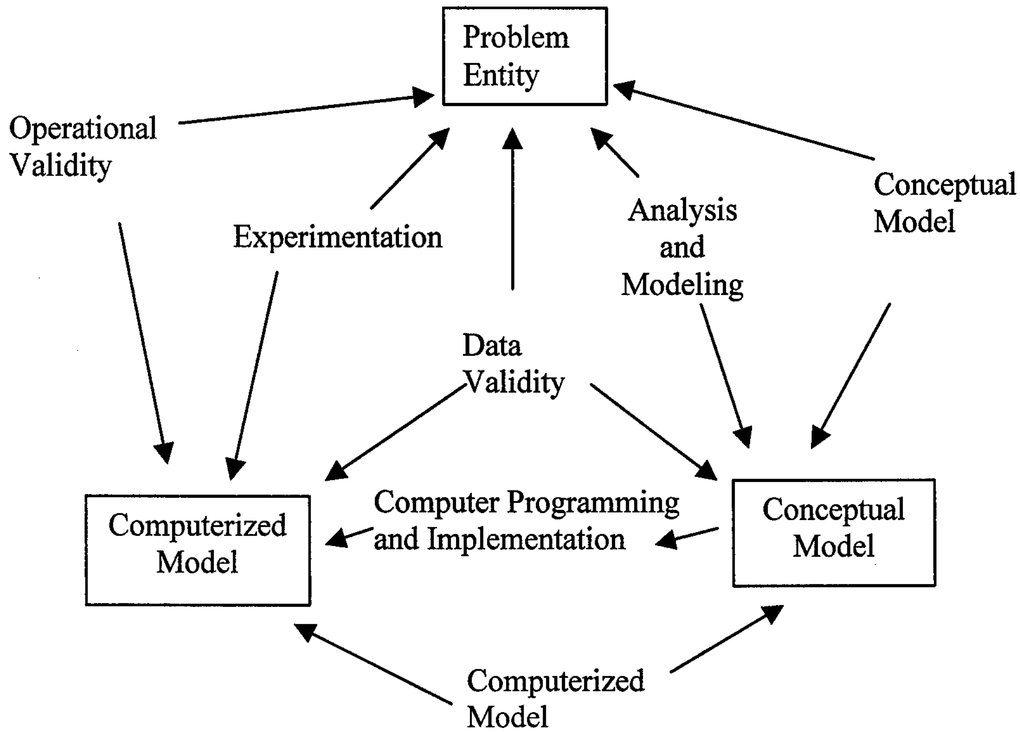

Figure 6. Simplified Version of the Modeling Process

The problem entity in Figure 6 is the system or the idea to be modeled. The conceptual model is the mathematical or logical representation of the problem entity. The computerized model is the conceptual model implemented on a computer.

Conceptual model validity is defined as determining that the theories and assumptions underlying the conceptual model are correct and the model representation of the problem entity is reasonable for the intended purpose of the model. Computerized model verification is defined as ensuring that the computer programming and implementation of the conceptual model is correct. Operational validity is defined as determining that the model's output behavior has sufficient accuracy for the model's intended purpose over the domain of the model's intended applicability. Data validity is defined as ensuring that the data necessary for model building, model evaluation and

testing, and conducting the model experiments to solve the problem are adequate and correct [15].

2.4.1 Verification

According to DoD directive 5000.59, verification is the process of determining that a model implementation accurately represents the developer's conceptual description and specifications [22]. In more general terms, verification is usually an iterative process that determines if a model and its resultant simulation accurately represents both what is required and what the modeling and simulation developer said would be built in accordance with those requirements. Verification ensures that the modeling and simulation will have more complete requirements, a better defined conceptual model, a more thorough and correct design, and cleaner implementation with fewer operational bugs. Since fewer things are left to chance, this means lower development risk, easier use and maintenance and a more satisfied user.

In general, computerized model verification ensures that the computer programming and implementation of the conceptual model are correct. The type of computer language used affects the probability of having a correct model program. The use of a special purpose simulation language generally will result in having fewer errors than using a general-purpose simulation language.

After the computer program has been developed, implemented and programming bugs removed, the program must be tested for correctness and accuracy. First, the simulation functions should be tested to see if they are correct. Straightforward tests can

be used here to determine if they are working properly. Next, each sub-model and the overall model should be tested to see if they are correct.

According to Sargent, there are two basic approaches to testing, static and dynamic testing [15]. In static testing, the computer program of the computerized model is analyzed to determine if it is correct by using techniques such as correctness proof, structured walk-through and examining the structure properties of the program. In dynamic testing the computerized model is executed under different conditions and the results are used to determine if the computer program and its implementations are correct. This includes both the values obtained during the program execution and the final values. There are three different strategies used in dynamic testing. The first one is the bottom-up testing, which means testing the sub-models first and then the overall model. Second one is top-down testing, which means testing the overall model first and then testing the sub-models. Third one is mixed testing. Mixed testing uses a combination of bottom-up and top-down testing.

The techniques commonly used in dynamic testing are traces, investigations of input- output relations, internal consistency checks, and reprogramming critical components to determine if the same results are obtained. We have to be careful while checking the correctness of the computer program and its implementation because the data, the conceptual model, the computer program or the computer implementation may cause errors.

According to Whitner and Balci, the verification effort should at least include the following requirements for it to be complete [21] :

1. Identification of the verification agent.

2. A description of the model or simulation version or release and developing organization.

3. Complete identification and description of the verification methodologies and activities, organizations, and individuals involved in the verification process from the beginning.

4. Results of the verification effort, including any identified modeling and simulation limitations.

2.4.2 Validation

Validation can be divided into different categories. Youngblood and Pace present two categories of validation methods [11]. First one is the conceptual model validation. The other one is implementation validation method. Conceptual validation is the review of assumptions, algorithms, modeling concepts, data availability, and architecture of the conceptual model. This determines if the model is expected to provide an acceptable representation of the subject for the intended application. Implementation (results) validation is the review process that compares model responses to known or expected behavior to determine that the responses are sufficiently accurate for the intended uses.

Ketanni and Oral divide validation into four parts, structural, experimental (results), operational, and data validation [12]. Structural validation's core concern is the degree of assumption and theoretical relevance underlying the formal model of the real world event. Experimental validation is concerned with the quality of solutions, the types of solutions, the nature of solution techniques, and the efficiency of solution procedures. Operational validation refers to the usability, usefulness, timeliness, and cost of

implementation of the model's recommended decision. Data validation involves the sufficiency, accuracy, appropriateness, availability, maintainability, reliability, and the cost of the data. Experimental validation is outside the scope of this effort because we are not evaluating the quality of the solutions. Other validation categories are explained in depth in the following section.

2.4.2.1 Conceptual, Operational, and Data Validation

Conceptual model validity can be defined as determining that the theories and assumptions underlying the conceptual model are correct and the model representation of the problem entity and the model's structure, logic, and mathematical relationships are reasonable for the intended purpose of the model. The theories and assumptions should be tested using mathematical analysis and statistical methods on problem entity data. Examples of assumptions are linearity and independence.

Next, each sub-model and the overall model must be evaluated to determine if they are reasonable and correct for the intended purpose of the model. Face validation techniques can be used at this point. If errors are found in the conceptual model, it must be revised and conceptual model validation should be performed again.

Operational validity is concerned with determining that the model's output behavior has the accuracy required for the model's intended purpose. This is where most of the validation testing and evaluation takes place. The computerized model is used in operational validity. And thus, any deficiencies found may be due to an inadequate conceptual model, an improperly programmed or implemented conceptual model or due to invalid data.

All of the validation techniques are applicable to operational validity. Which techniques and whether to use them objectively or subjectively must be decided by the model development team or the VV&A agent.

Even though data validity is usually not considered to be part of the model validation, it is very important. Data limitations are often the reason why attempts to validate a model fail due to inaccurate data resulting in incorrect outputs in the model. Generally, data are needed for three purposes: for building the conceptual model, for validating the model, and for performing experiments with the validated model. In the case of model validation, the agent needs conceptual data and data for validating the model of the intended purpose. This data can be difficult, time consuming, and costly to obtain.

To build a conceptual model we must have sufficient data on the problem entity to develop theories. These theories are used in building the model, to develop the mathematical and logical relationships in the model. In addition to this, behavioral data is needed on the problem entity to be used in the operational validity step. If these data are not available, high model confidence usually cannot be obtained, because sufficient operational validity cannot be achieved.

As mentioned earlier it is difficult, time consuming and costly to obtain accurate and appropriate data. Unfortunately, there is not much that can be done to ensure that the data are correct. The best that can be done is to develop good procedures for collecting and maintaining data. Test the collected data and screen for outliers and determine if they are correct. If the amount of data is large, a database should be developed and

maintained. If data are not available, high model confidence cannot be obtained. This situation directly impacts the credibility and the accreditation of the model.

2.4.2.2 Validation Techniques

The validation techniques can be used either subjectively or objectively. Generally, combinations of techniques are used for validating the sub-models and the overall model. According to Sargent, et al, all or some of the techniques can be used during the validation process [15]:

Animation: The model's operational behavior is displayed graphically as the model moves through time. The users can see the dynamic displays of the simulated system. These displays can be pictures, drawings, geometric shapes or even cartoons. Since the users are familiar with the real system, they can detect programming and conceptual errors.

Comparison to Other Models: Outputs of the simulation model are compared to the results of other valid models. The VV&A agent should make sure that; a credible authority has validated the other model. A credible model helps to increase the credibility and chance of accreditation of the model at hand. After the comparison is made, the results from both models should be evaluated by using graphical or statistical analysis techniques.

Degenerate Tests: The degeneracy of the model's behavior is tested by appropriate selection of values of the input and internal parameters. For example, does the average number in the queue of a single server continue to increase with respect to time when the arrival rate is larger than the service rate?

Event Validity: The events of occurrences of the simulation model are compared to those of the real system to determine if they are similar. For example, in a hospital simulation, deaths are inevitable occurrences when human beings are considered. Therefore the hospital simulation should generate deaths at the appropriate rate.

Extreme Condition Tests: The model structure and output should be plausible for any extreme or unlikely combination of factor levels in the system. This helps the user to identify that the model is operable in every region within its defined domain. An example for this case may be, if the process inventories are zero, then the production output should also be zero.

Face Validity: Face validity is asking people who are knowledgeable about the system if the model and its behavior are reasonable. These people should be subject matter experts. This technique can be used determining if the logic in the conceptual model is correct and if a model's input and output relationships are reasonable.

Fixed Values: Constants are used for various model input and internal variables and parameters. This allows the checking of model results against easily calculated values or analytical solutions. The results can also be checked with real life data if there is enough information about the real system.

Historical Data Validation: A portion of the historical data (or data collected on a system for building or testing the model), is used to build the model and the remaining part is used to test if the model behaves as the system does. This testing can be conducted by populating the simulation model with either sample from historical distributions or traces.

Historical Methods: There are three historical methods of validation; rationalism, empiricism, and positive economics. Rationalism assumes that everyone knows whether the underlying assumptions of the model are true. Logic deductions are used from these assumptions to develop the correct or valid model. Empiricism requires every assumption and outcome to be empirically validated. Positive economics requires only that the model be able to predict the future and is not concerned with a model's assumptions or structure. This can also be explained as casual relationship or mechanism theory.

Internal Validity: Several replications of a model are made to determine the amount of internal variability in the model. A high amount of variability, which is a sign of lack of consistency, may cause the model's results to be suspect and may also question the appropriateness of the policy or system being investigated.

Operational Graphics: Values of various performance measures may be shown graphically as the model moves through time. For example, the performance measure might be number of customers in a queue or percentage of servers busy and the dynamic behavior of them are visually displayed as the simulation model is running.

Parameter Variability-Sensitivity Analysis: This technique consists of changing the values of the input and internal parameters of a model to determine the effect upon the model's behavior and its output. The same relationship should occur in the model as in the real system. Those parameters that are sensitive, i.e., cause significant changes in the model's behavior or output, should be made sufficiently accurate prior to using the model.

Predictive Validation: The model is used to predict the system behavior and then comparisons are made between the system's behavior and the model's forecast to determine if they are the same. The system data may come from an operational system or from experiments performed on the system. An example for these is the field tests, which are common for military systems.

Traces: The behavior of different types of specific entities in the model is followed through the model to determine if the model's logic is correct and if the necessary accuracy is obtained.

Turing Tests: People who are knowledgeable about the operations of a system are asked if they can discriminate between system and model outputs. In other words, they compare performance of the system against that of an expert in the blind trials.

2.4.3 Accreditation

Model accreditation is specified for a particular modeling and simulation use or application. The accreditation decision evaluates the appropriateness of a particular model for a specific application. The verification and validation data provide the decision-maker with a basis for accepting the simulation outputs and integrating the results into the decision process. The accreditation decision documents the acceptance of the simulation results to support a specific test report. A model may require multiple accreditations to support different categories of uses and different configurations.

The modeling and simulation application is defined by the system to be simulated, the key variables to be studied, the test measures of interest, and the data collection requirements for the simulation. The definition of the system must identify the key

features of the system or environment that the model represents. A decision maker's comfort in the simulation results depends on the quality of the simulation outputs, the applicability of the simulation test results to the test measures, and the correspondence of the model components to the real world problem. This application specific information may not be recognizable from the verification and validation reports, but is essential to the accreditation decision.

The model developers have a complete understanding of the assumptions and limitations of their models, however, they may not have the same level of understanding of the intended application. Accreditation combines the application specific information with the model capabilities to evaluate the credibility of the model. Building model credibility is a continuous and iterative process. The first step extends from verification and validation to documenting the model status and capabilities. The second step is application specific and needs to be reviewed with each model use. The heart of the accreditation decision is the transfer of the model developer's confidence and understanding to the model user.

2.5 RAPTOR Overview

RAPTOR is a modeling framework developed by HQ AFOTEC to enhance existing analysis capabilities. It allows for quick creation of reliability, maintainability, and availability models for almost any system. The user model their systems graphically by drawing reliability block diagrams (RBD) and answering questions about the way components fail and are repaired. These component failure and repair rates can then be simulated to determine RM&A characteristic of the overall system.

A reliability block diagram is a graphical and mathematical representation of the relationships between a system's components (blocks) and their effect on the resulting system readiness level. RAPTOR uses RBDs as a foundation for simulation. Random numbers are drawn from the failure and repair distributions of each component, and then these failure and repair times are scheduled against a simulation clock. By flowing time through the system while failing and repairing each component, RAPTOR determines behavior of the overall system, such as its availability. A simple example of RBD is shown in Figure 7.

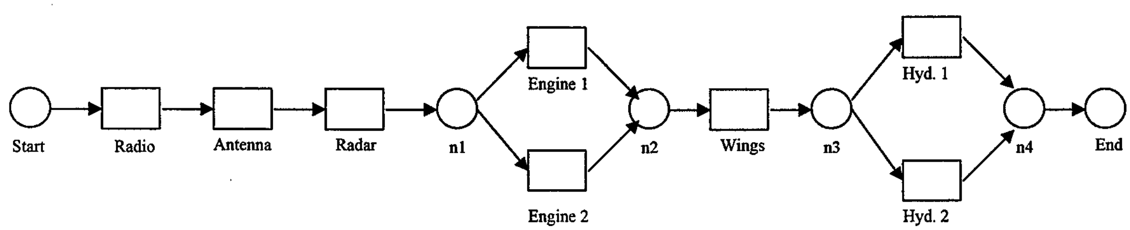

Figure 7. Sample RBD

Figure 7 shows a simple aircraft system. The aircraft has two engines and two hydraulic systems that work independently and at the same time. If one of the engines or the hydraulic systems does not function, the overall system is degraded but still functioning. This assumes node 2 and node 4 are defined as ½ nodes, which implies the system works as long as 1 out of 2 of the parallel components are working properly. If one of the other systems is not functioning then the overall system is not mission capable.

RAPTOR allows long-term analysis of reliability, maintainability, and availability performance as well as mission-based scenarios with specified time duration. It graphically depicts the changing conditions of the system, allowing the users to visually

examine the operating system during simulation [8]. This feature enhances model building, validation, and presentation capabilities. RAPTOR provides one of the most realistic representations of real world modeling in the field of reliability simulation.

2.6 Summary

The life cycle application of verification, validation, and accreditation is extremely important for successful completion of complex and large-scale modeling and simulation efforts. Applying the VV&A techniques throughout the life cycle of a model is time consuming and costly. Because of this time constraint, verification and validation are often sacrificed early in a modeling and simulation effort. This can impact the model accreditation.

The following should be taken into consideration while determining which verification and validation techniques should be selected for a particular model: model type, simulation type, and modeling and simulation objectives.

How much to test and when to stop testing depends on the modeling and simulation objectives. The testing should continue until sufficient confidence is achieved in acceptability of modeling and simulation results. The level of confidence is determined by the modeling and simulation objectives. Every new simulation project should be evaluated separately as a new and unique challenge. It is not possible to prove that a model is absolutely correct but proper VV&A techniques create confidence in a model for the results to be accepted and used in decision making.

3. Methodology

3.1 Adequate Verification, Validation, and Accreditation Plan

At the beginning of the verification, validation, and accreditation effort, the most significant task is the development of a plan that outlines all necessary activities. The appropriate level of planning is to identify the required verification and validation activities in each development phase. These activities provide the necessary evidence needed before continuing to the next phase of model building. The end result is a collection of evidence used in the validation and accreditation process.

Effective planning depends on having a comprehensive understanding of the development effort and what is expected from the VV&A effort. Other essential planning factors include selecting the most effective and practical tools, techniques, and methods to accomplish the necessary VV&A activities. A general VV&A plan should include the following types of information:

1. Purpose and description of the VV&A application.
2. Scope of the effort and VV&A responsibilities.
3. Identification of the Project Manager, user agencies, contractors, VV&A agent and responsibilities of each.
4. Intended use of the modeling and simulation.
5. The developer of the model (when, who etc.)
6. Accreditation agent and authority.
7. Descriptions of planned verification activities to be performed.
8. Descriptions of planned validation activities to be performed.

9. Descriptions of planned accreditation activities to be performed.

10. VV&A report

11. Descriptions and locations of VV&A archives to support accreditation and future reuse.

The first six steps are the general information about a model. The first step describes why and how the verification, validation, and accreditation will be accomplished. The second step identifies the scope of the VV&A effort. This is the general outline of the VV&A plan and it clarifies the boundaries of the effort. The third step identifies who the model project manager is, who is conducting the VV&A, and who is going to use the results. Step four identifies when, how, and for what purposes the modeling and simulation will be used. Step five identifies by whom and where the modeling and simulation was developed. Step six identifies the accreditation agent and the authority to accredit the model.

Up to step seven, all information can be easily gathered. In the next three sections steps seven, eight, and nine will be extensively discussed. Step ten and eleven can be accomplished after all other steps are accomplished and finalized.

3.2 Verification

The accuracy of transforming a problem formulation into a model specification or the accuracy of converting a model representation from a flowchart form into an executable computer program is evaluated in model verification. Model verification deals with building the model right. Model verification starts with verifying the requirements, which establish the starting point of the model. They show what the model

is capable of or not. They should be clearly stated at the beginning of the development process and should be consistent with each other.

The design verification is the next step in verification. This is the step where the model is being put together. After the model is developed, then it must be checked for errors in the code. Errors or bugs in the code will result in inaccurate results when the model is used. Different techniques can be used at this step as long as they are effective and useful for the purpose.

The steps of verification will be evaluated in the following order for this generic plan:

1. Requirements Verification.
2. Design Verification.
3. Implementation (Results) Verification. (Code checking will be explained briefly in this section.)

Animation is a general technique that can be used during the overall verification process. This technique will also be explained.

3.2.1 Requirements Verification

Requirements verification is one of the major parts of VV&A planning. Some requirements come from the user and the community that will be supported by the model. Others come from the interactions among the user, model, project manager, and the developer. These requirements specify how the developer plans to provide software and hardware model capabilities to meet the operational needs of the user. One of the VV&A tasks is to compare these requirements and determine the differences and disconnects between them. As this effort takes place, the VV&A agent learns more about the model

and this helps him to begin making up the VV&A plan. This initial effort focuses on identifying the accuracy, consistency, and completeness of the requirement definitions. The model requirements are the foundation of the entire model development process. They should be established clearly from the beginning and understood by all the participants of the development process.

During the requirements verification phase each requirement should be compared to its source for correctness and consistency. To ensure a requirement is being appropriately addressed in the simulation, the developers must find a way to detect or measure it. There are two types of risks associated with requirements verification phase. The first one is development risk, which shows the ability of the developer to implement the requirement without compromising the capabilities or the performance. The second one is the operational risks that are associated with using the model for its intended purpose.

Requirement verification should be accomplished as early as possible to ensure a proper foundation of the development process. As more detailed information becomes available or if changes occur that effect the requirements, requirement verification may have to be revisited throughout the development process. Any risk or uncertainty associated with the development of a new model and risks associated with specific VV&A tasks should be documented. Once the objectives and requirements have been defined, a risk assessment can be conducted to identify and prioritize the developmental and operational risks. The VV&A agent then should focus on the difficult issues of verifying individual requirements.

3.2.2 Design Verification

During the design phase, the developers take the requirements, algorithms, and interactions that are described in the conceptual model, and determine how to design them to support the software application. The purpose of design verification is to ensure that all these features are correctly and completely included in the design representations and documentation. The VV&A agent must ensure that all requirements are correctly traced during the design phase.

Design verification consists of different tasks. The first one is verifying the interfaces. All interfaces should be verified to ensure that information could be passed as needed for the intended use. Interfaces are the communication devices of the user and the model. If correct communication cannot be established, the model will not perform its intended purpose. The second task is about the amount of time or memory needed to complete a critical software process. The timing and sizing requirements for different processes should be checked to ensure that they do not interfere with the execution of the simulation. Next the key algorithms should be examined for their accuracy and appropriateness for the application. The algorithms are translated in to understandable formats so the logic and the accuracy can be analyzed. This analysis may involve rederiving the equations or evaluating the suitability of numerical techniques. It checks that algorithms are correct, appropriate, stable, and meet all accuracy, timing, and sizing requirements. Algorithm analysis examines the correctness of the equations and numerical techniques, truncation and rounding effects. Each design verification task should be documented by listing its objectives, assumptions, constraints, methods and

results. The results must be reported to the model project manager and the user for review and future reference.

All the tasks above rely on the products coming from the design process. These products include the design documentation and representations, algorithms, design reviews and walkthroughs, diagrams and drawings. The accuracy of all these products is important for the efficient completion of the design verification.

3.2.3 Implementation Verification

At this phase both the hardware and the software are constructed, tested and the actual data are installed and tested. Since the hardware is involved, the scope of the VV&A effort can grow significantly because integration of the hardware is included in the evaluation. At this point, the code should be run on static and dynamic analyzers to identify errors and to ensure accurate execution. Then, the interfaces between the components should be checked to ensure accurate relationships. Software connections and assignments to hardware components should be checked for appropriateness.

If there is a special hardware involved in the system (like weapon system models, aircraft motion simulators), the amount of verification required will significantly increase. Developmental and operational test plans and procedures should be reviewed to determine if they provide information needed for the VV&A effort. If possible, share the results of the testing activities to minimize costs and increase efficiency.

During the development phase the code is checked continuously in code walkthroughs, however an overall code check must be accomplished at the end of the development phase. Code check is explained in depth in the next section.

3.2.3.1 Checking the Code

After the model has been developed, implemented, and most of the bugs removed, the program must be tested for correctness and accuracy. First, the simulation functions should be tested to see if they are correct. Next, each sub-model and the overall model should be tested.

There are two testing approaches- static and dynamic testing. In static testing, the computerized model is analyzed to determine if it is correct by using correctness proofs, examining the structure properties of the program, and walkthroughs. A walkthrough is an informal meeting at which the program developer (or author) of a design or programming product explains the details of the product to the other members of the team. It is expected that the reviewing team members will assist the author in identifying errors or suspicious areas that would probably be undetected until some later time in the project life cycle. According to Deutsch, there are two types of walkthroughs [17].

1. Design Walkthroughs: An individual designer presents the design to other team members by explaining the structure charts, interfaces, and file definitions. Team members look for flaws in the design.
2. Code Walkthroughs: An individual programmer explains the structure and flow of a small section of the code to other team members. Team members again look for errors or bugs in the code.

Walkthroughs are intended to compensate for shortcomings in the formal design reviews. The primary objective of a walkthrough is to find errors. The success of walkthroughs in achieving the objective will vary depending on the team members and

their dedication to the task. Both design and code walkthroughs increase the probability of finding errors at an earlier time. There are also other advantages of walkthroughs:

1. Weak areas like efficiency and readability problems are illuminated.

2. The threat of a walkthrough improves the product quality of designers and programmers.

3. Junior designers and programmers learn techniques from their senior associates. 4. The marginal cost of conducting walkthrough is small in comparison to the cost of fixing errors discovered later in the product life cycle.

Types of walkthroughs can be further distinguished in terms of their formality and membership. This type of categorization is applicable to both design and code walkthrough [17].

1. Internal continuous walkthroughs within the team environment.

2. Milestone walkthroughs that may include other attendees as well as the team members.

Milestone walkthroughs are generally more effective. These walkthroughs are conducted when a particular product milestone has been achieved. Inclusion of knowledgeable participants from outside the team will help to broaden the objectivity of the review. The participants of a milestone walkthrough would include:

1. A manager
2. The author
3. Other team members
4. The system engineer

The manager coordinates the walkthrough and maintains order. Most of the time it is effective to have the system engineer present if he is the one who defined the requirements. The author is the most important person as the walkthrough continues. They explain the content and flow of the product to the other team members. He can use operational scenarios or test cases. The team members and other reviewers; if there are any, can make constructive comments on suspected problem areas. Interaction and questions are highly encouraged during the walkthrough. A record of all errors detected should be documented. This is to ensure the errors are corrected at a later time. Walkthroughs are relatively inexpensive devices to identify existing design or programming errors before they become permanent in the product.

The modeler also needs to read the code to ensure that the right data and the logic have been entered. A useful idea is to get a second, independent check on the code. On the other hand, as an alternate, the code might be expressed in a non-technical format and a non-expert could check the data and the logic. This can be especially useful for obtaining the system expert's opinion.

3.2.4 Visual Checks and Animation

The visual display of the model is a very powerful aid for verification. By running the model and then watching how each element behaves, the model logic is compared against the real world behavior. According to Robinson, some of the techniques that can be used are [26]:

1. Stepping through the model event by event.

2. Stopping the model while it is running and then predicting what will happen next and after that running the model on and checking what happens.

3. Setting up interactive conditions on purpose to force certain events to take place.

4. Closing or isolating some areas of the model so it runs faster, and reducing the time to perform verification.

5. Explaining the model to those people who are knowledgeable or experts about the real system in an attempt to get their opinion about how close the model is to the real system.

6. Tracing the progress of an item through the model.

It is also very useful to watch a model running for a period of time. Just by watching the model run, a lot can be learned about the behavior of the model. Another useful technique would be to demonstrate the model, either formally or informally, to those who have a detailed knowledge of the system. This is helpful in two ways. First, this helps the VV&A agent to identify any shortcomings in the model. Second, involving the real system experts to this effort should increase the credibility of the work.

The actual and the expected results from a simulation run are compared using the output reports. One valuable report is a trace of a simulation as it moves through time. This is a step-by-step history of every event, which took place during the run and is written to a file. Inspecting this report can help identify and rectify any problems before they become permanent flaws in the model. An inspection begins with the distribution of the item to be inspected. Participants should analyze the item on their own. During the inspection, each item is jointly analyzed to find as many errors as possible. All errors found are recorded, but no attempt is necessary to correct the errors at this time.

However, at some point in future, it must be verified that all errors documented have been corrected.

The VV&A agent may also use animation to verify the model if the system being modeled is dynamic. Animation can be an effective way to find invalid model assumptions and to enhance the credibility of a simulation model. Moving pictures or cartoons are used as dynamic displays of the simulated system so that the user can visualize the simulation. Since the users are familiar with the real system, they can detect programming errors. For example in a traffic simulation, cars might cross through each other, or in a bank simulation, customers may disappear during the simulation run. Since all these problems are not the intentions of the programmer, they are the concerns of verification. There are also risks of using animation. Generally, animations make the simulations run slower. Due to this handicap, users tend to concentrate on short simulation runs so the problems that occur only in long runs may go unnoticed. It is the VV&A analyst's job during verification to continue the runs long enough to create the rare events. Rare events occur infrequently and only after a certain time unit in the simulation. If the simulation is not run for enough time units, any problems related with these rare events may go unnoticed.

3.3 Validation

The basic principle of validation is controlled execution of a program with known inputs and outputs combined with internal measurement of the behavior of the simulation model. The notion of controlled execution forms the basis for a systematic methodology for carrying out the program validation process for a simulation model system. A

systematic methodology to accomplish the testing gives the user confidence in the quality of the end product. Each step in the systematic methodology involves repeated applications of the basic testing principles.

Some test situations will involve a number of separate tests of individual modules in the model. One of the objectives of testing is to construct scenarios that exercise a program's structure in a particular way. It is also important to develop series of tests that affirm the actual content of the simulation model. Using analytical results of the simple system or actual results from known or prior systems are examples of such tests.

Another important aspect of validation is the sensitivity of input parameters. The simulation is run under a variety of input settings and checked if the output is reasonable. The most definitive test of a simulation model's validity is to establish that it's output data closely resemble what is expected from the actual system. The output data are compared to those from the existing system itself. If the two sets of data compare "closely", then the model of the existing system is considered "valid". The greater the commonality between the existing and proposed systems, the greater our confidence in the model of the proposed system.

Once the VV&A agent proves that the simulation model is programmed correctly, he can then move to the next level where they have to answer the question: is the conceptual simulation model an accurate representation of the system under study or the real world? The validation process shows the faithfulness of the model or simulation to the thing being represented. It provides crucial evidence to support a model or simulation's credibility for a particular application.

The validity of a model or simulation also reduces risks. This is especially critical in a project that depends on simulation for its development or management decisions. The validity of a model or simulation supports, but does not guarantee, project credibility in all cases. Most of the time, credibility of a model relies upon the trust that the user has in the validation results, the process that produced that results, and the people that executed that process. Users must believe in the credibility of a model or simulation before they will use it. Validation of a model or simulation starts with complete and consistent statement of the requirements derived from the user's objectives. Description of the characteristics of the model and description of the expert knowledge against which the model will be compared for accuracy and exactness should also be included.

Validation process will be evaluated in three parts. The first part will be validation with available data. Comparison to real life system technique will be explained in this section. The second part will be validation with no data. Comparison to other models and comparison to analytical results are the two techniques that will be explained in this section. In the last part the usability of the model will be evaluated.

3.3.1 Comparison with Real System

3.3.1.1 Data Validation

Historic or expected data collected from the real system can be compared to the results of the simulation when it is run under the same conditions. To obtain a valid model, the VV&A agent should try to measure the inputs and outputs of the real system. In the real world, data are available in different quantities.

Sometimes it is difficult or even impossible to get relevant data. For example, if the simulation studies of a nuclear war are taken in to consideration, it is impossible to

get the necessary data. For a simulation study of animal population dynamics, it might be a major problem to obtain data on the animal behavior. The analyst or the agent might end up spending more effort on data collection than the study itself. In most cases, some data is available. In the case of an existing manufacturing system, the agent can obtain realistic data about the system.

The military commonly conducts field tests in order to obtain data on future systems. In some applications, there is an overload of data. This generally happens when the data are obtained electronically. For example, in the simulation of the performance of computer systems, data on the system-state can be collected each nanosecond. In addition, the analysts may use the model history to improve validation based on the additional data. In the end, real world data may be very little or abundant. Moreover, the data may show observation error, which complicates the comparison of real and simulated results.

Validation data are actual measurements from the real world or are best guess information provided by subject matter experts. They are used in validation to determine if the results of the simulation are correct for the simulation to be useful for the intended purpose. Validation data are the real world facts used for comparison to validate the results of a simulation.

3.3.1.2 Comparison Techniques

When the VV&A agent succeeds in obtaining data on real system, it is fed into the model in historical order. After running the simulation program, the VV&A agent obtains a time series of simulation output and compares that time series with the historical time series for the output of the existing system. While conducting validation

the VV&A agent should not sample from a distribution of real world input values. The historical input values must be used in historical order. After the agent has validated the simulation model, he can compare different scenarios using sampled inputs. The VV&A agent can compare a time series of simulation model output with a historical time series output by using several different techniques.

The output data of the real system and the simulated system can be plotted in a diagram such that the horizontal axis denotes time and the vertical axis denotes the real and simulated values respectively. The user may eyeball the time paths to decide whether the simulation model accurately reflects the system of interest, or in other words, determines if the model's output behavior has sufficient accuracy for its intended purpose. Different types of graphs like histograms, box plots, or scatter plots can be used. A variety of graphs using different types of measures such as the mean, variance, maximum, distribution, and time series of variable are required. It is important that appropriate measures and relationships should be used in validating the model. The graphs can also be used in the face validity technique where experts make judgments on whether a model possesses sufficient accuracy for its intended purpose. Finally, the graphs can be used in Turing tests.

In Turing tests, the agent presents a mixture of simulated and real time series to their clients, and challenges them to identify the data that were generated by the computer. The clients may correctly identify some of the data by chance, however, the agent can test this coincidence statistically. In addition, the VV&A agent can directly use mathematical statistics to obtain quantitative data about the quality of the simulation model. The simulation output data form time series whereas elementary statistical

procedures use identically and independently distributed (i.i.d.) observations. Nevertheless i.i.d. observations can be derived from the simulation so that elementary statistical theory can be applied.

Kleijnen applies the statistical theory in the following example [19]. Let w_i and v_i denote the average waiting time on day i in the simulation and the real system respectively. Suppose that n days are simulated and observed in reality, $i = 1, \dots, n$. The averages do not need to be computed from a steady state time series of individual waiting times. They may be calculated from the individual waiting times of all customers arriving at a specific time interval. Each day includes a start up, and a transient phase. In this case, both the simulated averages (w_i) and the real averages (v_i) are i.i.d. If the historical arrival and service times are used to drive the simulation model, then the n paired differences ($d_i = w_i - v_i$), are statistically i.i.d. The t statistics given below can be used to check the output accuracy:

$$t_{(n-1)} = (\bar{d} - \delta) / (s_d / \sqrt{n}) \qquad (3.1)$$

where $\bar{d}$ denotes the average of the n number of d's, δ is the expected value of d, and s_d represents the estimated standard deviation of d. The variable $d_i = w_i - v_i$ is the difference between simulated and real average waiting time on day i when using the same arrival and service times. Hence $\bar{d}$ is the average of the n differences between the n average simulated and n average real waiting times per day.

Suppose that the null hypothesis is H_0: $\delta = 0$, and gives a value t_{n-1} that is significant for $|t_{n-1}| > t_{n-1;\alpha/2}$. The simulation model is rejected, since the model gives average waiting times per day that deviate significantly from reality. In the case of a

non-significant $|t_{n-1}|$, the conclusion is that the simulated and the real means are statistically the same and the simulation is valid. This interpretation is a good one but the simulation is only a model, so δ is never exactly zero. The bigger the sample size is, the smaller the critical value $t_{n-1;\alpha/2}$ is; for example, for a fixed $\alpha = 0.05$ and n = 5 and n = 121 respectively, $t_{n-1;\alpha/2} = 2.776$ and $t_{n-1;\alpha/2} = 1.98$ respectively. All other things being equal, a simulation model has a higher chance of being rejected as its sample size gets bigger. Simulating many days gives a precise estimate $\bar{d}$ and hence a significant t_{n-1}. In a case like this, model misspecification leads to rejection if the sample size n were infinite. If the sample is very large, then the t statistic is nearly always significant for $\delta \neq 0$. Nevertheless, the simulated and the real means may be compared to determine if the simulation is valid enough. For example, $E(w_i) = 1000$ and $E(v_i) = 1001$ ($\delta = 1$), then the simulation model is good enough for all practical purposes.

In general, when testing the validity of a model through statistics, the VV&A agent can make either a type I or a type II error. That is, they may reject the model while the model is valid; type I or α error. On the other hand, they may accept the model while the model is not valid; type II or β error. The probability of a β error is the complement of the power of the test, which is the probability of rejecting the model when the model is wrong. The probability of a type I error in simulation is called model builder's risk; the type II error probability is the model user's risk.

The power of the test of H_0: $\delta = 0$ increases as the model specification (true δ) increases. A significance or critical level α means that the type I error probability equals α. The probability of a β error increases as α decreases, given a fixed number simulated days. If α is kept constant and n increases, then $t_{n-1;\alpha/2}$ decreases. Another problem to

overcome is the selection of a value for α. Popular values are 0.10 and 0.05 [27]. Theoretically, the VV&A agent should determine these values by accounting for the financial consequences or overall impact of making type I and type II errors respectively.

Tests based on classic regression analysis can be used. The first test looks at the relative value of the output variable. If we suppose on a certain day (let it be 4) the real average waiting time is relatively high, higher than expected ($v_4 > E(v)$), it seems reasonable to require that on that day the simulated average is also relatively high ($w_4 > E(w)$). The test checks that v and w are positively correlated: H_0: $\rho > 0$ (ρ denotes the linear correlation coefficient). The VV&A agent can formulate a validation test, which is simulated, and real responses do not necessarily have the same mean, but they are positively correlated. To investigate this correlation, the agent may plot the n pairs (v_i, w_i). The graphical approach can be formalized through the use of the ordinary least squares algorithm. Testing the hypothesis of positively correlated v and w is simple if v and w are bivariate normally distributed. This is a realistic assumption because of the central limit theorem. It can be proved that such a bivariate normal distribution implies a linear relationship between the conditional mean of one variable and the value of the other variable:

$$E(w \mid v = v.) = \beta_0 + \beta_1 * v \qquad (3.2)$$

The agent can use ordinary least squares to estimate the intercept and the slope of the straight line that passes through the cloud of points (v_i, w_i). The test concerns one-sided hypothesis H_0: $\beta_1 \leq 0$. The t statistic can be used to test this null-hypothesis. This

test means that the agent rejects the null-hypothesis and accepts the simulation model if there is strong evidence that the simulated and the real responses are positively correlated. This test must be modified if the simulation is meant to predict absolute responses not relative responses corresponding to different scenarios. In this case, the agent should formulate the test that the means of w (the simulated response) and v (the historical response) are identical, and if a historical observation exceeds its mean, then the corresponding simulated observation tends to exceed its mean too. These two conditions lead to the composite hypothesis H_0: $\beta_0 = 0$ and $\beta_1 = 1$, which implies $E(w) = E(v)$. To test this composite hypothesis, the agent should compute the Sum of Squared Errors (SSE) with and without that hypothesis (which corresponds to the reduced and the full regression model respectively), and compare these two values. If the resulting F statistics is significantly high, the agent should reject the hypothesis and conclude that the simulation model is not valid.

3.3.1.3 Sensitivity Analysis

Models and sub-models with unobservable inputs and outputs cannot be subjected to tests defined in the comparison techniques. In this case, the agent can apply sensitivity analysis in order to determine whether the model's behavior agrees with the judgments of the experts. Since the experts can predict what to expect from the system for different settings of inputs, sensitivity analysis results can be compared to these expectations. If the systems are simple enough, outputs can be predicted using simple analytic methods. If the outputs are available for different kinds of settings, sensitivity analysis is a very useful tool.

Sensitivity analysis or in other words, what-if analysis can be defined as the systematic investigation of the reaction of model outputs to intentional changes in model inputs or model structure. For example, in the case of a queuing simulation model what happens to the output when the arrival rate doubles or decreases to half? Experts can predicted before hand what to expect in both cases and then the simulation can be run with different inputs. The outputs should be similar to what the experts are predicting. Designs of experiments and regression analysis are the most common techniques that are used for sensitivity analysis. Most of the time practitioners apply an inferior design of experiments. They only change one simulation input at a time. If compared to fractional factorial designs (such as 2^{k-p} designs), the one at a time designs give estimated effect of input changes that have higher variances. Not only they are less accurate, these designs cannot estimate interactions among inputs.

The results of these experiments with simulation models can be analyzed and used for interpolation and extrapolation. The VV&A agent can plot the simulation output (let it be y) versus the simulation input (let it be x_k), one plot for each input k with k=1, … ,K This practice can be formalized through regression analysis. If we let y_i denote the simulation output in run *i* of the K simulation inputs, with $i = 1, \dots, n$, n denotes the total number of simulation runs. Furthermore, let x_{ik} be the value of simulation input k in run *i* and β_k the main or first order effect of input k, $\beta_{kk'}$ the interaction between inputs k and k', and e_i the approximation (fitting) error in run *i*. Then the input –output behavior of the simulation model may be approximated through the regression model as Kleijnen states [19]:

$$y_i = \beta_0 + \sum_{k=1}^{K} \beta_k \chi_{ik} + \sum_{k=1}^{K-1} \sum_{k'=k+1}^{K} \beta_{kk'} \chi_{ik} \chi_{ik'} + e_i \qquad (3.3)$$

The validity of this approximation must also be tested by using cross validation. Cross validation uses a subset of simulation inputs and output data to get estimated regression parameters ($\hat{\beta}$). Then it employs the estimated regression model to compute the forecast $\hat{y}$ for some other input combinations. The comparison of the forecasted output $\hat{y}$ and simulated output y is used to validate the regression model. This approach gives estimates, $\hat{\beta}$ of the effects of the various inputs. These estimated effects should have the right signs. Wrong signs show computer errors or conceptual errors. If there are any sensitivity estimates with wrong signs, the simulation model needs to be corrected.

Classical experimental designs with $n > K$ may require excess computer time especially when the simulation study is still in its early phase. A screening technique based on sequential experimentation with the simulation model can be used. The aggregated inputs can be split up until finally the important individual inputs are identified and their effects are estimated. In some cases, it is remarkable that this statistical technique identifies some inputs, which were originally thought to be unimportant. The magnitude of the sensitivity estimates show which inputs are important. For important inputs the VV&A agent should try to collect data on the input values that may occur in practice. If the data can be collected for these inputs, then the comparison techniques discussed in the previous section can be applied.

Before executing the experimental design (either one at a time or fractional factorial), the agent must determine the experimental domain. The experimental domain is "the limited set of circumstances under which the real system is to be observed or

experimented with" [19]. The design will tell us how to explore the experimental domain, because the model may be valid in one experimental domain but invalid in another one. For example, older data may not be representative of the current system. Different types of laws might have ruled the old system. Similarly, a model is accurate only if the values of its input data remain within a certain range. There are some objections to this idea because some experts think that a model should maintain valid under extreme conditions. This is not a very hard problem to overcome because the definition of extreme is relative to each expert. The only problem is to figure out how they defined extreme.

For this research we will comply with the idea that the simulation model is valid within a certain area of its inputs only. This area can be defined as the K dimensional hypercube formed by the K input ranges and within that area the simulation model's input-output behavior might vary [28].

As a conclusion the sensitivity analysis should be applied to identify important inputs and test for realistic output behavior. This information is useful even if there are many data on the input and output of the simulated system. Collecting information on the important inputs is worth the effort and might be useful for future research.

3.3.2 Validation with Limited Experimental Data or No Data

Sometimes the cost and the time required to conduct experimentation may limit the amount of experimental data available for validation. In this situation, using experimental or empirical data collected from other activities for similar situations can be helpful. Another option is extending the range of experimental data with calculations from consistent and validated models or inputs from subject matter experts.

The process of constructing referents needs as many credible sources as possible. Lack of experimental data requires reliance on other sources. Multiple data sets from different sources can be merged into a single referent. The consistency of each set against each other must be checked very carefully to improve the credibility of the referent and the validation. Sometimes, even a single experimental point can support validation. But, this cannot be done without credible data from other sources. Just by itself, a single experimental data point cannot validate a model or simulation for any purpose. In general the VV&A agent should use extreme caution when extrapolating from a limited number of field tests to assess overall model validity.

3.3.2.1 Comparison with Other Models

Comparison with other models is particularly useful when no real system data are available. However, real data should be used when available. Indeed, using other model comparisons in addition to the real world data comparison can increase the confidence on the model.

One approach is to compare the simulation model against a mathematical model. It is generally not common for a mathematical model to be able to predict the outcome of the simulation exactly. Otherwise the simulation model would not have been built. However, a mathematical model may be able to give a crude approximation of the outputs of the real system. Examples of mathematical models that might be used are paper calculations, spreadsheet analysis and queuing theory. It is sometimes useful to simplify the simulation model to the extent that a mathematical model can predict exactly the outcome of the model. A specific example of this can be the use of deterministic

models. This is a simulation model from which all the random events have been removed. In many cases it is possible to determine mathematically the exact outcome of such a model.

Comparisons can also be made against other simulation models of the same or similar systems. For instance, a more detailed model of the system may have been developed for some other purpose. Or a general simulation model for similar systems is already available. If another simulation model is used during comparison it is supposed that the other simulation model is valid and credible.

3.3.2.2 Comparison to Analytical Results

Analytical comparison techniques examine the structure of models and simulations. There should be enough notional data or symbolic values that permit the agent to compare the model or simulation with the calculated analytical results. This comparison technique must be used starting from the development phase of the model, especially during the conceptual model validation.

Generally, notional values are preferred rather than actual data because they are easier to obtain or assign. These symbolic values are fed into the model and outputs are produced. They are also transformed analytically and results from these analytical transformations are compared to the result obtained from the simulation model. The results may not exactly coincide but they should be in agreement at a specified level of confidence.

Analytical comparison has some disadvantages. As the model or the system that is simulated grows in size or becomes more complex, the calculations required for comparisons also grow in size. Some parts of the system that are simulated may not be

represented with symbolic values. In this case these parts may be evaluated independently. Even though complexity of the system creates difficulties for the agent, using symbolic values tests the basic characteristics of the simulation model.

3.3.3 Usability of the Model

Even though usability of the model does not seem to be an important subject on the developer side, it is very important for the user. The VV&A agent should make sure that the model is as user friendly as possible. In most cases, being user friendly increases the popularity and the credibility of the modeling and simulation product.

First of all, the VV&A agent should spend enough time to validate the interface. Each and every capability of the interface should be used. Every feature of the model should be tested in order to comply with the requirements. For the end product to be free of errors, all features must respond in the way that they are programmed. A surprised user will not increase the credibility of the model. In addition, these kinds of problems will require additional verification and validation efforts before the accreditation phase. Additional verification and validation will yield to additional time requirements, and time is a very valuable constraint.

3.4 Accreditation

According to DoD 5000.59, accreditation is required for all important applications where models and simulations are employed [14]. The primary purpose of accreditation is to establish modeling and simulation credibility. The accreditation agent accumulates data that supports modeling and simulation suitability for a given application. Although

the entire VV&A process contributes to model credibility, the accreditation assessment and the resulting accreditation decision are the key steps in establishing model suitability for a specific application. Accreditation is always associated with a specific purpose or application. It is a comparison of a model's capabilities and attributes with the modeling requirements generated by the specific problem. To make a judgment about model suitability, the agent must have clear description of what the model can or cannot do. Two critical steps for accreditation are to clearly define the problem and modeling requirements associated with it and to obtain a clear understanding of the model's proven capabilities and limitations. Comparison of the modeling and simulation requirements with basic model information leads to an accreditation decision. Three outcomes are possible; use an existing model as is, modify an existing model, or build a new model. This is referred to as choosing a modeling approach.

Once the modeling approach has been chosen, one builds a body of evidence that shows the selected model or simulation is acceptable for use in the intended application. This evidence generally consists of basic information about the model or simulation, validation and verification results, and information about the data. This information generally answers five questions:

1. What does the model do?
2. How good is the software?
3. Are the model outputs realistic?
4. Can the model be operated properly and the results interpreted correctly?
5. Is the data satisfactory?

The answers to these questions are the basis for judging model acceptability. While building the body of evidence, the agent uses all the available information generated during model development or by the past model users. If there are any gaps in this information, the information package is incomplete for the accreditation purpose. In this case, additional verification, validation or other data collection efforts are necessary. Prior to accrediting the model or simulation, the agent conducts the necessary verification and validation to fill in the gaps. The supplemental verification and validation information, along with existing information, is used to conduct the accreditation assessment. At this step, the requirements of the problem are compared against the capabilities and characteristics of the model and the suitability is determined. This is the last step, which results in the accreditation recommendation.

3.4.1 Assessment

Once the accreditation requirements (modeling and credibility) are known, and the VV&A information have been gathered, the agent can conduct the accreditation assessment. The accreditation assessment is essentially a comparison of a model's capabilities and attributes with the modeling and simulation requirements. The comparison is usually an iterative process. The first iteration usually leads to a list of information deficiencies about the model and these information deficiencies can lead to supplemental verification and validation work. Once the supplemental verification and validation information has been gathered, a final comparison is made to generate the basis that supports accreditation. In some cases this assessment can uncover model deficiencies with respect to the requirements of the problem.

If model deficiencies are identified during this assessment, the VV&A agent or team determines whether the deficiency is tolerable. However, in some cases the deficiency is intolerable, especially when there is a better modeling technique that avoids the deficiency. Manual adjustments of input or output values, or changes to parameters within the model may often compensate for model deficiencies. Adjustments can preserve the ability to use a particular model with minor deficiencies. Another option is to limit the model's use to certain scenarios where the outputs are known to be acceptable. The last step in assessment is good reporting. The agent or the team of agents assemble the findings and make an overall assessment about model suitability and risks of using the model reviewed. Any recommendations or additional verification and validation work should also be included.

3.4.2 Accreditation Report

One of the responsibilities of the accreditation agent is to prepare a good report of the accreditation assessment. The accreditation report serves as a checklist to ensure the accreditation assessment or process has generated the necessary information. The essential elements to include in the accreditation report are [18]:

1. List of simulation acceptability criteria.
2. A description of simulation capabilities and limitations.
3. Summary report of the accreditation assessment showing how the simulation meets the acceptability criteria or if does not meet it, what kind of risks might be associated with the limitations of the simulation.
4. Accreditation statement.

All these elements can be contained in a single report or in multiple documents. The VV&A agent should ensure the user recognizes the importance of archiving this information. The agent should help the user to develop appropriate techniques to capture this information and ensure they have adequate resources for preserving it. The accreditation assessment report helps the VV&A agent or the team to focus on the suitability of the application instead of its capabilities. This helps the team to expedite clarifying the major issues and saves time.

The acceptability criteria (first element of the accreditation report) are documented as a separate report or included in the accreditation report. A description of requirements derived from the basic problem objectives and parameters are included. This action allows others to review and validate these requirements if necessary. The description of simulation capabilities, assumptions, and limitations related to the intended application is normally included as a part of the accreditation. All simulation assumptions and limitations identified during the development and associated verification and validation efforts are documented. The conceptual model, which is developed as a part of the new model development process, contains most of the simulation capabilities.

The accreditation assessment report is an essential document needed by the user to make the accreditation decision. For the accreditation of a new simulation model, the document presents evidence that the simulation model meets the acceptability criteria. If there are any criteria that are not met, the accreditation assessment report includes an evaluation of the impact of not meeting these criteria. In addition, the report lists the potential ways to fix these problems and the associated risks with them. This evaluation prioritizes the tasks again and distributes the resources objectively to meet the simulation

acceptability criteria. The accreditation assessment report includes comments about the evaluation and recommendations about the credibility and the accuracy of the data being used. If it is appropriate, this report includes the suitability evaluation of the operators necessary to properly run the simulation and interpret its results.

The typical accreditation statement is a brief (most of the time just one page) executive summary that includes a synopsis of the basis for the accreditation recommendation. This brief statement includes a list of limitations and recommended constraints on the accreditation, and an approval statement for the accreditation authority to sign. By itself, the accreditation statement only shows that an accreditation assessment has been completed. However, when this statement is contained in a package accompanied by other supporting documents, the entire package presents the logical basis for accreditation.

However, for the accreditation of a model, not all the steps in the generic VV&A plan have to be accomplished. Some of the steps can be omitted if applicable. In some cases additional steps might be added according to the accreditation acceptability criteria.

4. Application

As stated in thesis outline (Section 1.5), this chapter is going to be the application of the generic VV&A plan outlined in Chapter 3 to RAPTOR. At first, our intention was to apply only the validation portion of this generic plan. Validation only, however, is not adequate to understand model performance. Validation shows if a simulation model is an accurate representation of the real world or in other words if we have built the right model. As a matter of fact, we cannot be completely sure if we have built the right model unless we are sure that the model is working correctly. In order to be sure that the model is working correctly, a partial verification of the model should be accomplished before continuing with the validation effort. The verification of the model is outside the scope of this effort but an accreditation report with recommendations will be given in Chapter 5.

The verification should start with the verification of the distributions and random number generators used in the simulation model. These features have already been verified in the verification efforts of Raptor 2.0 (30 January 1996). The new version uses the same distributions and the random number generators. The inspection of trace reports technique is used to fulfill the partial verification requirement. This report is a trace of all simulation activities. The report is a step-by-step history of every event (event log) that takes place during the simulation run. This report is then written to a file and will be used to make the necessary calculations during verification.

While accomplishing the verification of RAPTOR 5.0, three different systems will be used as a foundation: a simple system, a complex system, and a system using the phasing feature of RAPTOR 5.0 (phasing is a new feature added to 5.0). The objective of

tracing these three systems is to use as many features as possible. Using one system, which captures all features would be difficult to analyze. More importantly, breaking the trace verification into three systems allows additional analysis to be accomplished. In a single system, it would be very hard to distinguish any possible errors and identify the source at the coding error. All three systems use different features and in the case of any errors, it is easier to identify and isolate the coding error. Another reason using three different systems is the time constraint. A complicated system takes more time to be simulated and also to be investigated. Breaking the system in to manageable sizes is more efficient.

Section 3.1 outlined the VV&A plan to be used for RAPTOR. The first six steps of this plan are explained in the next paragraph. Step 7(planned verification activities) and step 8(planned validation activities) will be accomplished in the following sections and the reports (step 10) will be given after these activities are performed. Step 11(description and location of VV&A archives to support accreditation and future use) is this total effort itself.

The purpose of this VV&A application is to validate RAPTOR simulation model for use by AFOTEC. This application is not a complete verification and validation effort. The VV&A agent's responsibility is to develop a generic plan and perform the intended V&V activities, and make recommendations for accreditation. This model will be used by AFOTEC for operational test and evaluation of Air Force systems. The model is developed by AFOTEC (both military and civilian contractors).

4.1 Verification

The test platform that is used during the verification is a pentium III, 450 Mhz computer with 128 MB RAM. All simulation runs are accomplished on this platform.

Verifying the requirements is the first step in verification. Older versions of RAPTOR did not have features like phasing, event blocks or cost analysis. Evaluation of these new features has priority over other features. In the verification part, all these features are tested to see if they are operating correctly.

The design of the model has already been determined. This VV&A effort is being conducted after the simulation has developed. A few recommendations are made. Another important issue is checking the code. Code checking requires extended amount of time and an expertise in ModSim simulation language used to create RAPTOR. This is beyond the scope of this research.

Therefore only implementation (results) verification from the generic plan is accomplished. Three different systems are used in this part of verification. The first system used is a simple system. This system uses the basic features of RAPTOR. The second system is complex system that uses most features of RAPTOR. The third system uses the phasing feature of RAPTOR. Phasing is a new feature added to the new version of RAPTOR.

4.1.1 Simple System

The first part of results verification will be done using the simple system shown in Figure 8.

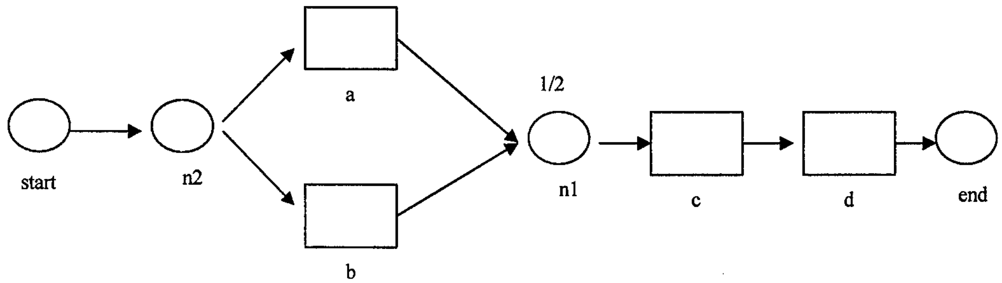

Figure 8. Simple System Built in RAPTOR.

This example is a simple system consisting of four nodes and four blocks. With this system, the basic features of RAPTOR will be tested for accuracy and correctness. The system will be run for 1000 time units and 10 replications.

The first node is the start node. n2 is the second node in the system and n1 is the third node. The last node in the system is the end node.

The distributions and other special features of blocks are given in Table 1 below:

Table 1. Distributions of Figure 8.

Block	**Failure Rate**	**Repair Rate**	**Spare Policy**	**Initial Level For Spares**	**Spare Arrival Policy**	**Avg. Logis. Delay**	**System Dep.**	**Initial Block/ Spare Cost**	**Oper./ Repair Cost**
a	Expo 90	Expo 10	Custom	2	1 Per 100 Units	30 Units	No	1/1	1/1
b	Expo 90	Expo 10	Custom	2	Emerg. 1 per 24 Units	None	Yes	1/1	1/1
c	Expo 80	Expo 20	Series Spares Pool	100	_	None	No	1/1	1/1
d	Expo 70	Expo 30	Series Spares Pool	100	_	None	No	1/1	1/1

The event log of this simulation is given in Appendix A. This event log was used to track all the failures, capture down-times, and analyze output parameters. Each item on the results table is calculated manually and then compared to end of simulation results. The individual failure and repair data is tracked through the entire event log. The mathematical calculations are done using Microsoft Excel.

End Of Run Report for Simple System.rbd Simulation

End of Run #1

Ending Simulation Time	= 1000.000000
Number of system failures	= 14
Number of component failures	= 37

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	**29.493177**	0.759034	97.380220	29.886280
Down Times	**41.935394**	1.667713	126.136214	40.818164
Total Time Between Component Failures	26.833212	0.807927	106.356761	23.274074
Component Repair Times	23.912194	0.135673	93.067539	24.064833

Availability	= **0.412904**
MTBM	= **11.159581**
GreenTime	= **26.878026 %**
YellowTime	= **14.412423 %**
RedTime	= **58.709552 %**

The meaning of each item and how they are calculated are explained in detail below according to RAPTOR User's Manual [8].

Mean Time Between Downing Events: This is the average time between events, which bring the entire system down. MTBDE is calculated as the total time the system operates (in a green or yellow condition) divided by the number of downing events. The total time the system operated without any problems (green time) and degraded (yellow) was calculated manually and they are 268.780257 and 144.124230 respectively. And there were 14 failures in the system. In our case MTBDE is equal to:

$$(268.780257+144.124228) / 14 = \mathbf{29.493177}$$

Mean Down Time: This is the average amount of time the entire system is down. It is calculated as the total amount of system down time divided by the number of downing events. The total time the system is down is equal to red time. In our case MDT is equal to:

$$587.095515 / 14 = \mathbf{41.935394}$$

Availability: This is the ratio of the time the system is up to all time. It reflects what percentage of time the system is available for use. It is calculated as MTBDE divided by the sum of MTBDE and MDT. RAPTOR calculates availability as the sum of green and yellow times divided by the sum of green, yellow, and red times. In our case it is equal to:

$$(268.780257 + 144.124228) / 1000 = \mathbf{0.412904}$$

Mean Time Between Maintenance: This is the average amount of time between any maintenance actions performed on any components of the system. It is calculated as the total time the system operates that is divided by the total number of failures of all

components. The total time between maintenance events were tracked manually and summed 412.904485. MTBM is calculated by dividing this amount by the total number of failures (37).

In our case it is:

412.904485 / 37 = **11.159581**

Green Time: Green time is the percent time during a simulation when no components in the RBD are failed. In our system this time is **26.878026 %.**

Yellow Time: Yellow time is the percent time during a simulation when some components in the RBD are failed, but the overall system is not down. In our case this is **14.412423 %.**

Red Time: Red time is the percent time during a simulation when some components on the critical path in the RBD are failed, causing the overall system to be down. In our case this time comes up to **58.709552 %.**

The cost report (given below) gives the operation, repair, and spares costs of each block and the overall system relative to the costs defined in the simulation. The initial cost, operation cost, repair cost, and spare cost per each unit can be defined and entered as an assigned value to the simulation. The overall cost of operating the system is calculated automatically at the end of the simulation run. The cost of each and every unit in the system can also be calculated. This gives the user the flexibility of changing any high cost item in the system with a lower costing item.

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
a	1.00	642.59	80.23	12.00	735.83
b	1.00	370.46	41.77	4.00	417.23
c	1.00	614.99	385.01	0.00	1001.00
d	1.00	679.75	320.25	0.00	1001.00
Block Totals	4.00	2307.80	827.27	16.00	3155.06

Pool	InitCost	RecurringCost	Total
Series	100.00	0.00	100.00
Pool Totals	100.00	0.00	100.00

Grand Total 3255.06

In the simple system all the costs related with each block and the overall system have been calculated according to the event log and matched the cost report.

4.1.2 Complex System

The second system used in the verification is a more complex system. Different features like spare pool, cold standby pool, resource pool, average logistics delay are used in this system. The system in Figure 9 is configured of 13 blocks and 11 nodes including the start and end node. The system runs for 1000 time units and 10 replications. The distributions and features of each block and are given in Table 2. All the nodes act the same as in the simple system. The event log of this simulation is given in Appendix B. From this event log every item on the end of run results report can be calculated.

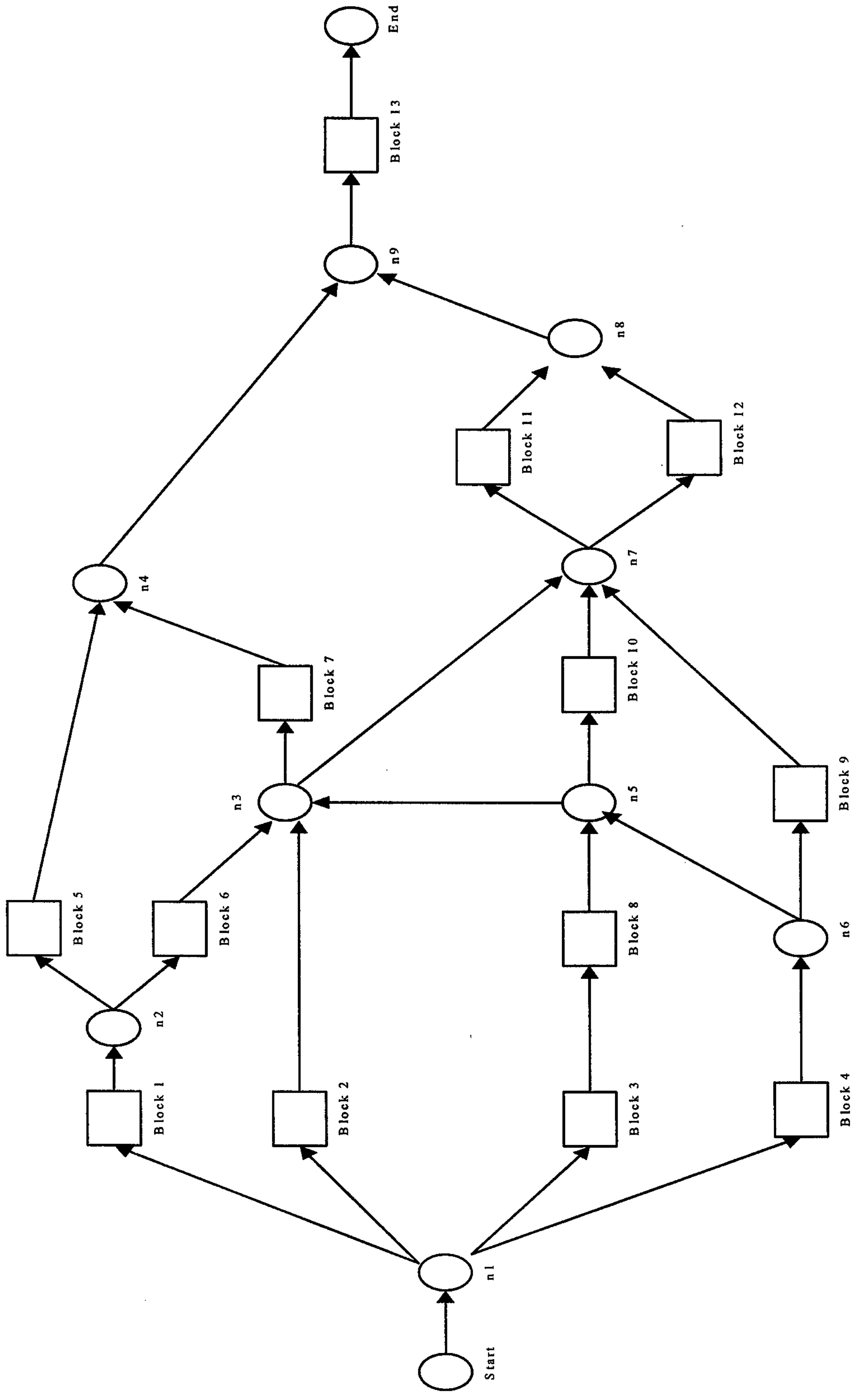

Figure 9. Complex System Built in RAPTOR.

Table 2. Distributions for Complex System.

Block	Failure Rate	Repair Rate	Spare Policy	Initial Level For Spares	Spare Arrival Policy	Average Logistics Delay	System Dependent	Auto Switch Prob.	Initial Level For Pool	Oper/ Repair Cost	Initial/ Spare Cost
1	Expo 90	Expo 10	Infinite	_	_	None	No	_	_	1/1	1/1
2	Expo 80	Expo 20	Sp1 Spares Pool	10	1 per 500 Units	None	No	_	10	1/1	1/1
3	Expo 65	Expo 35	Csp1 Cold Stby Pool	_	_	None	No	1 in 10	5	1/1	1/1
4	Expo 50	Expo 50	Rp1 Resource Pool	Infinite	_	None	No	_	10	1/1	1/1
5	Expo 70	Expo 30	Sp2 Spares Pool	5	Emerg. 1000 Units	None	Yes	_	5	1/1	1/1
6	Expo 75	Expo 25	Csp1 Cold Stby Pool	_	_	None	No	1 in 5	5	1/1	1/1
7	Expo 85	Expo 15	Sp1 Spares Pool	10	1 per 500 Units	10	No	_	10	1/1	1/1
8	Expo 30	Expo 70	Sp2 Spares Pool	5	Emerg. 1000 Units	None	Yes	_	5	1/1	1/1
9	Expo 35	Expo 65	Infinite	_	_	None	No	_	_	1/1	1/1
10	Expo 60	Expo 40	Sp1 Spares Pool	10	1 per 500 Units	None	No	_	10	1/1	1/1
11	Expo 70	Expo 30	Sp2 Spares Pool	5	Emerg. 1000 Units	None	No	_	5	1/1	1/1
12	Expo 80	Expo 20	Sp1 Spares Pool	10	_	None	No	_	10	1/1	1/1
13	Expo 90	Expo 10	Infinite	_	_	None	No	_	_	1/1	1/1

End Of Run Report for Complex System.rbd Simulation

End of Run #1

Ending Simulation Time = 1000.000000
Number of system failures = 7
Number of component failures = 88

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	**59.723037**	5.225078	221.991348	70.239232
Down Times	**83.134106**	2.039983	499.458498	170.225402
Total Time Between Component Failures	11.021286	0.039612	48.036386	10.294478
Component Repair Times	39.202524	0.112041	217.157591	50.030583

Availability = **0.418061**
MTBM = **4.750696**
GreenTime = **0.094971%**
YellowTime = **41.711155%**
RedTime = **58.193874%**

The definitions of all features have already been made in the simple system. For the complex system only calculations will be shown for each feature.

Mean time Between Downing Events:

$$418.061261 / 7 = \mathbf{59.723037}$$

Mean Down Times:

$$581.938739 / 7 = \mathbf{83.134106}$$

Availability:

$$(0.949713 + 417.111548) / 1000 = \mathbf{0.418061}$$

Mean Time Between Maintenance:

$$418.061259 / 88 = \mathbf{4.750696}$$

Green Time: **0.949713**

Yellow Time: **417.111548**

Red Time: **581.938739**

The cost report (given below) gives the operation, repair, and spares costs of each block and the overall system relative to the costs defined in the simulation.

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
block4	1.00	312.77	687.23	10.00	1011.00
block3	1.00	880.00	621.24	0.00	1502.24
block2	1.00	351.70	13.80	0.00	366.49
block1	1.00	822.16	177.84	17.00	1018.00
block5	1.00	227.81	53.01	0.00	281.82
block6	1.00	945.00	300.82	0.00	1246.82
block7	1.00	265.11	64.16	0.00	330.27
block8	1.00	22.42	217.16	0.00	240.57
block10	1.00	315.55	174.81	0.00	491.36
block9	1.00	259.94	740.06	7.00	1008.00
block11	1.00	130.64	83.27	0.00	214.92
block12	1.00	418.74	86.35	0.00	506.09
block13	1.00	950.55	49.45	8.00	1009.00
Block Totals	13.00	5902.38	3269.21	42.00	9226.59

Pool	InitCost	RecurringCost	Total
sp1	10.00	2.00	12.00
sp2	5.00	3.00	8.00
Csp1	5.00	N/A	5.00
Rp1	N/A	687.23	687.23
Pool Totals	20.00	692.23	712.23

Grand Total 9938.82

In the complex system all the costs related with each block and the overall system were calculated according to the event log and all the results related with the cost report are correct.

4.1.3 System with Phasing

The third system used in the verification is a system with phasing. Phasing is a new feature added to version 5.0. Systems go through different stress levels during their lives. For example, the stress level for an air-conditioning unit is less in winter. Phasing feature helps us to model different failure rates or degradation factors in a system for the same simulation run.

The first step of evaluating the phasing feature is to show that the system changes its properties in each phase. To evaluate this feature, the system was run for 5000 time units with different phases starting at each 1000 time units. Statistics were collected for each phase. These statistics proved that the failure rate changed in proportion to the assigned degradation factor for each phase.

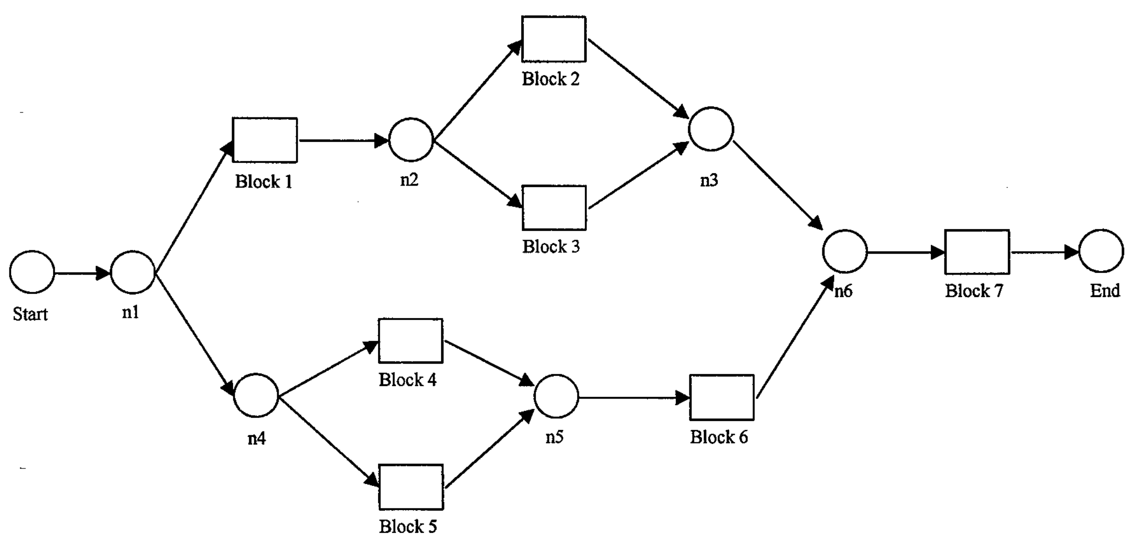

Figure 10. System with Phasing Built in RAPTOR

The system in Figure 10 is configured of 7 blocks and 8 nodes including the start and end node. The system will be run for 1000 time units and 10 replications. The distributions and features of each block are given in Table 3 below. All the nodes act the same as in the simple system. The event log of this simulation is given in Appendix C. From this event log every item on the end of run results report can be calculated.

Table 3. Distributions for System with Phasing.

Block	Failure Rate	Repair Rate	Spare Policy	Initial Level For Spares	Spare Arrival Policy	Degradation Factor	System Dependent	Oper/ Repair Cost	Initial/ Spare Cost
1	Expo 50	Expo 50	Infinite	_	_	5	No	1/1	1/1
2	Expo 80	Expo 60	Spares Pool	10	1 per 200 Units	10	No	1/1	1/1
3	Expo 50	Expo 25	Infinite	_	_	15	No	1/1	1/1
4	Expo 35	Expo 50	Spares Pool	10	1 per 200 Units	1	No	1/1	1/1
5	Expo 30	Expo 70	Infinite	_	_	3	No	1/1	1/1
6	Expo 50	Expo 15	Infinite	_	_	5	No	1/1	1/1
7	Expo 15	Expo 15	Spares Pool	10	1 per 200 Units	10	No	1/1	1/1

End Of Run Report for Phasing.rbd Simulation

End of Run #1

Ending Simulation Time = 1000.000000
Number of system failures = 4
Number of component failures = 24

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	**238.844341**	18.639987	49.982651	12.241958
Down Times	**11.155659**	2.365845	18.822031	6.146190
Total Time Between Component Failures	40.034307	0.407020	201.681831	48.885836
Component Repair Times	39.504491	5.936601	155.112565	38.720333

Availability = **0.955377**
MTBM = **39.807390**
GreenTime = **25.570548 %**
YellowTime = **69.967188 %**
RedTime = **4.462264 %**

The definitions of all features have already been made in the simple system. For the system with phasing only calculations will be shown for each feature.

Mean time Between Downing Events:

$$955.377362 / 4 = \mathbf{238.844341}$$

Mean Down Times:

$$44.622638 / 4 = \mathbf{11.155659}$$

Availability:

$$(255.705482 + 699.67188) / 1000 = \mathbf{0.955377}$$

Mean Time Between Maintenance:

$$955.377362 / 24 = \mathbf{39.807390}$$

Green Time: **255.705482**

Yellow Time: **699.671880**

Red Time: **44.622638**

The cost report (given below) gives the operation, repair, and spares costs of each block and the overall system relative to the costs defined in the simulation.

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
block1	1.00	945.44	54.56	2.00	1003.00
block2	1.00	982.77	17.23	0.00	1001.00
block3	1.00	963.20	36.80	1.00	1002.00
block4	1.00	339.87	515.33	0.00	856.21
block5	1.00	824.53	175.47	3.00	1004.00
block6	1.00	918.64	81.36	3.00	1004.00
block7	1.00	972.15	27.85	0.00	1001.00
Block Totals	7.00	5946.60	908.60	9.00	6871.21

Pool	InitCost	RecurringCost	Total
Spares Pool	10.00	5.00	15.00
Pool Totals	10.00	5.00	15.00

Grand Total 6886.21

In the system with phasing all the costs related with each block and the overall system has been calculated according to the event log and all the results related with the cost report are correct.

4.2 Validation of the Model

In the previous chapter it was pointed out that it is generally hard to find reliable data about systems. Since data about a real life system was not available, comparison to other models and comparison to analytical results are the two techniques used in validation process. General usability of the model is also evaluated under validation. This is necessary to show that the model is performing its intended purposes.

4.2.1 Comparison with Other Models

There are other valid reliability models that are used by different companies and agencies. These models are similar to one another but are different based on features, flexibility and capabilities. Three reliability models researched are AvSim, Relex, and Item. AvSim was selected and is used in our comparison. The same systems are created in both software packages and the results are compared for validation of our model. The first system that will be used is shown in Figure 11.

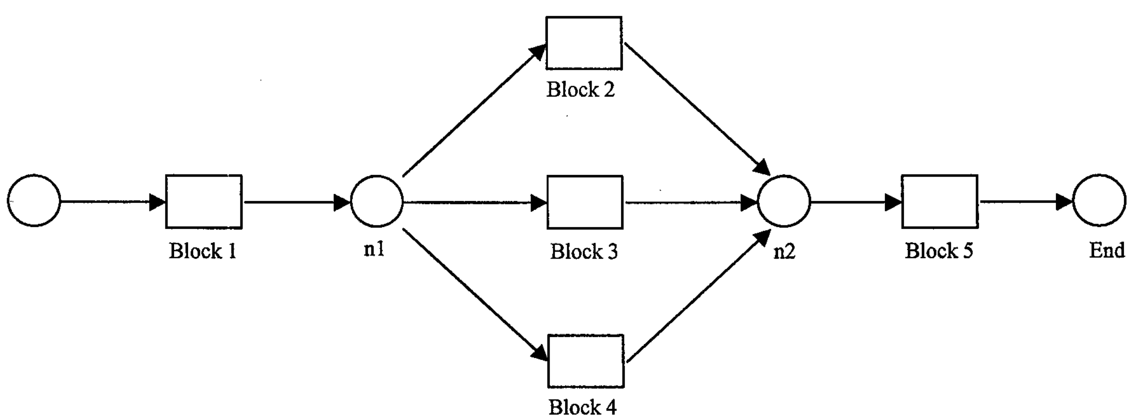

Figure 11. Sample System 1 for Comparison

Table 4. Sample System Features.

Block	Failure Rate Mean	Repair Rate Mean/ St.Dev.	Spare Policy	Oper/ Repair Cost	Initial/ Spare Cost
1	Expo 100	Lognor 10/2	Infinite	1/1	1/1
2	Expo 90	Lognor 20/1	Infinite	1/1	1/1
3	Expo 70	Lognor 25/1	Infinite	1/1	1/1
4	Expo 65	Lognor 15/3	Infinite	1/1	1/1
5	Expo 85	Lognor 5/1	Infinite	1/1	1/1

Results for Avsim Simulation:

Total Down Time Over Lifetime (TDT)	: 138.2
Mean Unavailability Over Lifetime (Q_m)	: 0.1401
Point Unavailability at Lifetime (Q)	: 0.2
Expected Number of Outages Over Lifetime (W)	: 19.1
Probability of 1 or More Outages Over Lifetime (F)	: 1
Mean Time to First Outage (MTTO)	: 47.38
Mean Time Between Outages (MTBO)	: 49.61
Mean Time to Restore to Service (MTTR)	: 7.16

Results for RAPTOR Simulation:

Availability (A_0)	: 0.8529
Mean Time Between Downing Events (MTBDE)	:47.5621
Mean Down Time (MDT)	: 7.7072
Mean Time Between Maintenance (MTBM)	:16.7727
Mean Repair Time (MRT)	:14.9241
Green Time	: 42.7700
Yellow Time	: 42.5235
Red Time	: 14.7064
System Failures	: 19.1

Although the output names may be different, the information indicated is very similar for both models. A test of hypothesis can be made to find out if there is any difference between the results of the two models for the simulation of the same system. Both simulations are run for 10 times so our sample size is 10. The first result that will be evaluated is availability. The sample standard deviation for AvSim simulation is $S_1 = 0.01047$ and the sample standard deviation for RAPTOR $S_2 = 0.03523$. The mean availability for AvSim simulation is $\mu_1 = 0.8599$ and the mean availability for RAPTOR simulation is $\mu_2 = 0.8529$.

The null hypothesis is: $H_0: \mu_1 - \mu_2 = 0$

Alternative hypothesis is: $H_1: \mu_1 - \mu_2 \neq 0$

The tests statistics that can be used to test this hypothesis is:

$$t_o = ((\bar{x}_1 - \bar{x}_2) - 0) / S_p\sqrt{((1 / n_1) + (1 / n_2))} \qquad (4.1)$$

Where

$$(S_p)^2 = ((n_1 - 1)(S_1)^2 + (n_2 - 1)(S_2)^2 / (n_1 + n_2 - 2) \qquad (4.2)$$

$$(S_p)^2 = ((10 - 1)(0.03523)^2 + (10 - 1)(0.01047)^2) / (10 + 10 - 2)$$

$$(S_p)^2 = 0.000675$$

$$S_p = 0.02598$$

Then the test statistics is:

$$t_o = (0.8599 - 0.8529) / 0.02598(\sqrt{((1 / 10) + (1 / 10))}$$

$$t_o = 0.60248$$

The rejection region for the hypothesis is:

$$t_o > t_{\alpha/2,\, n1+n2-2} \qquad (4.3)$$

For $\alpha = 0.05$ $t_{0.025,\,18} = 2.101$, so $t_o = 0.602$ is not greater than 2.101 and it is not in the rejection region. At 0.05 level of significance, we do not have the evidence to conclude that the two results differ from each other.

The second result that will be evaluated is mean time between downing events or mean time between outages. The sample standard deviation for MTBO is $S_1 = 8.1973$. The sample standard deviation for MTBDE is $S_2 = 13.3344$. The mean MTBO is $\mu_1 = 49.61$. The mean MTBDE is $\mu_2 = 47.5621$.

The null hypothesis is: H_0: $\mu_1 - \mu_2 = 0$

Alternative hypothesis is: H_1: $\mu_1 - \mu_2 \neq 0$

$$(S_p)^2 = ((n_1 - 1)(S_1)^2 + (n_2 - 1)(S_2)^2 / (n_1 + n_2 - 2)$$

$$(S_p)^2 = ((10 - 1)(8.1973)^2 + (10 - 1)(13.3344)^2) / (10 + 10 - 2)$$

$$(S_p)^2 = 122.5$$

$$S_p = 11.068$$

Then the test statistics is:

$$t_o = (49.61 - 47.5621) / 11.068(\sqrt{}((1 / 10) + (1 / 10))$$

$$t_o = 0.414$$

The rejection region for the hypothesis is:

$$t_o > t_{\alpha/2,\, n1+n2-2}$$

For $\alpha = 0.05$ $t_{0.025, 18} = 2.101$, so $t_o = 0.414$ is not greater than 2.101 and it is not in the rejection region. At 0.05 level of significance, we do not have the evidence to conclude that the two results differ from each other.

The third result that will be evaluated is mean time to restore to service or mean down time. The sample standard deviation for MTTR is $S_1 = 0.7903$. The sample standard deviation for MDT is $S_2 = 0.8949$. The mean MTTR is $\mu_1 = 7.16$ and the mean MDT is $\mu_2 = 7.7072$.

The null hypothesis is: H_0: $\mu_1 - \mu_2 = 0$

Alternative hypothesis is: H_1: $\mu_1 - \mu_2 \neq 0$

$$(S_p)^2 = ((n_1 - 1)(S_1)^2 + (n_2 - 1)(S_2)^2 / (n_1 + n_2 - 2)$$

$$(S_p)^2 = ((10 - 1)(0.7903)^2 + (10 - 1)(0.8949)^2) / (10 + 10 - 2)$$

$$(S_p)^2 = 0.7127$$

$$S_p = 0.8442$$

Then the test statistics is:

$$t_o = (7.7072 - 7.16) / 0.8442(\sqrt{((1 / 10) + (1 / 10))}$$

$$t_o = 1.45$$

The rejection region for the hypothesis is:

$$t_o > t_{\alpha/2,\, n1 + n2 - 2}$$

For $\alpha = 0.05$ $t_{0.025, 18} = 2.101$, so $t_o = 1.45$ is not greater than 2.101 and it is not in the rejection region. At 0.05 level of significance, we do not have the evidence to conclude that the two results differ from each other.

Two more systems are simulated to compare the results. The detailed results log for these simulations for both RAPTOR and AvSim are attached in Appendix D. A less detailed table will be used to compare these two system simulations. All assumptions made for the first system applies to the second and the third system.

The null hypothesis is: $H_0: \mu_1 - \mu_2 = 0$

Alternative hypothesis is: $H_1: \mu_1 - \mu_2 \neq 0$

Table 5. Test Statistics for System 2.

	μ_1	μ_2	S_1	S_2	S_p	t_o	$t_{0.025, 18}$
Availability	0.8216	0.8489	0.0448	0.0391	0.042	1.4534	2.101
MTBO MTBDE	100.207	85.3458	30.6357	17.7718	25.0437	1.3269	2.101
MTTR MDT	14.683	14.4298	0.5107	1.554	1.1566	0.4895	2.101
Number of System Failures	12.2	10.4	3.2249	2.2449	2.7784	1.4486	2.101

Table 6. Test Statistics for System 3.

	μ_1	μ_2	S_1	S_2	S_p	t_o	$t_{0.025, 18}$
Availability	o.9612	0.95	0.021	0.0144	0.018	1.3913	2.101
MTBO MTBDE	150.881	114.337	51.6158	22.1187	39.7078	2.0579	2.101
MTTR MDT	5.3068	5.7819	2.0127	1.407	1.7364	0.6118	2.101
Number of System Failures	7.1	8.6	2.4698	1.562	2.0663	1.6232	2.101

For both systems test statistics $t_o < t_{0.025, 18} = 2.101$ and it is not in the rejection region. At 0.05 level of significance, we do not have the evidence to conclude that the two results differ from each other.

As mentioned earlier these software packages have a lot in common, but they are not applicable for every system. The types of results obtained from the other two software packages do not match with RAPTOR output. The software packages are tracking different parameters. They are also very complicated to use which makes them hard to learn. This is one of the superiorities of RAPTOR. It is easy to learn and user-friendly.

RAPTOR results also contain the minimum value, maximum value, and the standard deviation for each item on the results box. The results for the third system are evaluated manually. There are no errors found on these calculations. The standard deviations are calculated manually and they are correct. The minimum and the maximum values in Appendix D are highlighted for easier evaluation.

4.2.2 Comparison to Analytical Results

Sample reliability system in Figure 12 is used to compare RAPTOR results with calculated analytical results. These are relatively simpler systems compared to the ones used in comparison to other simulation models. The system in Figure 12 will be evaluated in details. Other evaluations are based on the assumptions used in the first system.

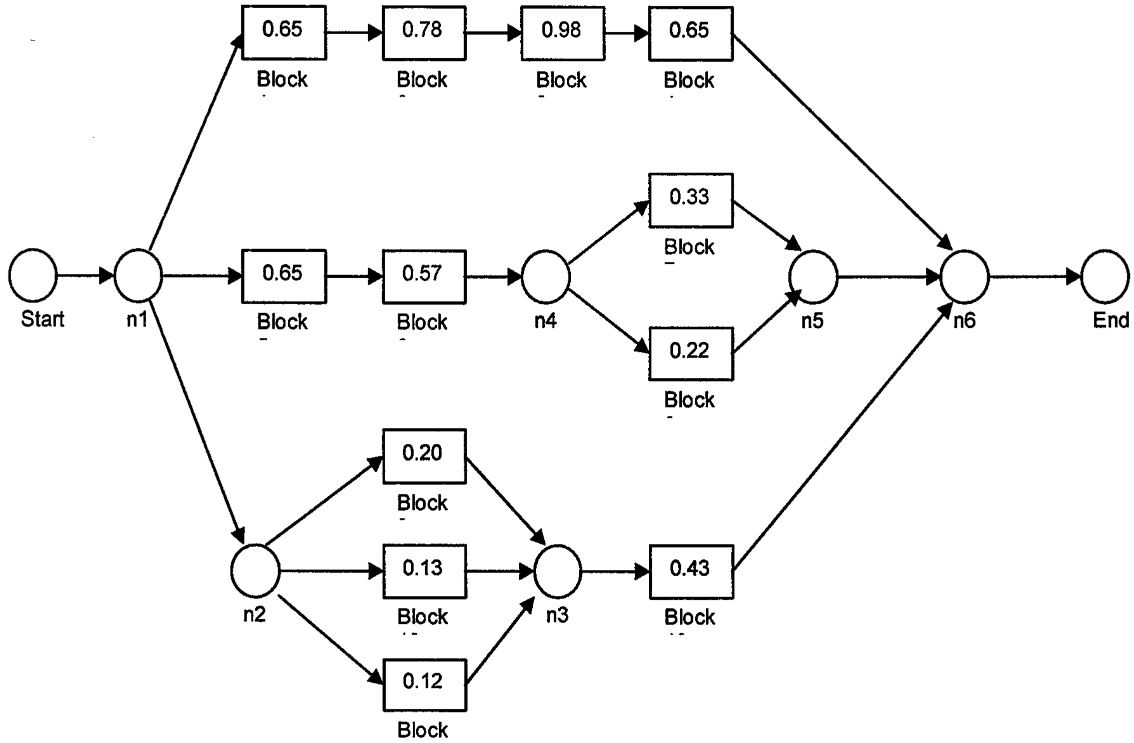

Figure 12. Sample RBD for Analytical Comparison.

Each block is assigned a certain reliability value. The event feature of RAPTOR is used in these comparisons. The event block is assigned certain probability values, while block diagrams are assigned distributions. The result obtained from event diagrams is 0.5386. The system's reliability obtained from analytical evaluations is $R_1 = p_1 = 0.5355$. Reliability obtained from block diagrams (using exponential distribution) for 10 runs is $R_2 = p_2 = 0.5360$. The standard deviation is $S = 0.06454$.

The null hypothesis is: $H_0: p_1 = p_2$

Alternative hypothesis is: $H_1: p_1 \neq p_2$

The test statistics that can be used is:

$$t = (p_2 - p_1) / (S / \sqrt{n}) \qquad (4.4)$$

The rejection region is:

$$|t| > t_{\alpha/2\,;\,(n-1)} \quad \text{where } t_{0.025\,;\,9} = 2.262$$

$$t = (0.5360 - 0.5355) / (0.06454 / \sqrt{10})$$

$$t = 0.0245$$

Since $|t| < t_{0.025\,;\,9} = 2.262$, and it is not in the rejection region, we can conclude that at $\alpha = 0.05$ level of significance we do not have enough evidence to reject H_0 and conclude there is no difference between the two results.

Two more simulation systems are also used in the evaluation. These systems are not described in detail. They are also built using the exponential distribution for failure and repair rates. The results for these two simulations along with other features are given in the following table. Comparisons are made using the same test statistics and rejection region. Assumptions used in the first comparison apply to these comparisons, too.

Table 7. Test Statistics for Analytical Comparison.

	Analytical Results	RAPTOR Results	Standard Deviation	t	$t_{0.025\,;\,9}$
Simulation 1	0.5355	0.5360	0.06454	0.0245	2.262
Simulation 2	0.8968	0.9051	0.03529	0.7437	2.262
Simulation 3	0.9992	0.9991	0.00197	0.1605	2.262

For both systems test statistics $t_o < t_{0.025,9} = 2.262$ and it is not in the rejection region. At 0.05 level of significance, we do not have the evidence to conclude that the two results for both simulations differ from each other.

4.2.3 Usability of the Model

In general terms, RAPTOR is fairly easy to learn and very user-friendly. Most of the features are working correctly and without any deficiencies. However, the following comments are provided addressing RAPTOR's usability and possible improvements. There are a few minor problems that should be fixed to improve this modeling tool.

1. The first problem is the position of the information box that appears during the simulation runs. This box appears at the lower left portion of the interface and hinders the lower bar that gives the number of replications during the simulation run. This box can be moved to the top and it changes to a bar when moved, but it does not stay there in the next run. Each time the simulation begins, the box appears at the bottom left. This box should be moved to the top where it does not restrict the view and should stay there at the end of every simulation run.
2. During the creation of the simulation system the editing function is hindered by the links between the blocks. While selecting a part of the simulation, if the selected portion is within two grid squares of a link, the model only selects the link instead of the whole selection area. It is not possible to select and cut or select and copy a portion of the system if the

user is close to the links. This is because even though the links are just lines, they occupy much more than their volume.

3. If the modeler is using a mouse with a scroll button and tries to select an area with the scroll button the zoom setting of the whole systems resets to 3% automatically. This only happens if a blank place on the interface (a place not including any blocks, nodes, etc.) is chosen.

4. If the modeler tries to select an area including RBDs with the scroll button, the software gives " The exception: Access Violation occurred in the MODSIM Debugger" error. An attempt to continue after this error message causes the program to shut down without any warning and without the ability to save any necessary information. Right after the error message the zoom on the interface also resets to 3%.

5. While linking blocks and nodes at the bottom of the screen, if the mouse cursor is close to the bottom edge, two more links appear with a 30° angle to the original one protruding from the same origin. They tend to go down to the bottom page without an end. These two extra links disappear when the page is scrolled up or down.

6. All other features including File (New, Open, Save, Save As, Simulate, Print, Print Selection, Print Setup, Close, Exit), Edit (Add, Select All, Cut, Copy, Paste, Clear, Details, Block Defaults, Mass Edit), Options (View Tables, Pools, Phases, Preferences) and the sub-features of these features work without any error. The sub-features under File are used during the routine execution of simulation runs. The sub-features under Edit are used

during the creation of the systems. The sub-features under options were used during the creation of different systems. Tables were viewed at the end of every simulation run. Pools were created in some of the system. The phasing feature was used in the system with phasing. All options under preferences are tested and they work correctly.

7. The Help feature also works correctly but there is no help file in the model. There is also no online help file available. This makes it hard and unsatisfying in some cases. The phasing feature is a new addition to RAPTOR 5.0. Even a modeler who is comfortable with the old versions might need some help to fully understand all the special features about phasing. It is definitely a must that a help file is included to the program.
8. While minimizing the whole interface, the interface minimizes shifting right. This makes the minimize, restore, and close buttons unreachable unless the whole interface is dragged into viewable area of the monitor.

5. Conclusions and Recommendations

This chapter is the final accreditation report for RAPTOR. The model results must match the known results for the model to be accepted as credible. RAPTOR results showed a precise resemblance to known results and therefore no statistical differences were found between these results.

5.1 Verification and Validation Assessment of RAPTOR

The Rapid Availability Prototyping for Testing Operational Readiness (RAPTOR) is an easily usable and important simulation modeling tool for reliability, maintainability, and availability analysis of all AFOTEC systems. It has the ability to help mitigate common operational test and evaluation problems such as high cost of testing, inadequate time and resources. These are very important issues that need precise and correct solutions. The only way to obtain these precise and correct results is to have an accredited working simulation model. Otherwise operational ability of the Air Force would be degraded and this would inversely influence the combat outcomes.

Verification and validation of the model assures the user or the accreditation authority that the right model is built and it is working correctly. It would be very assuring if the simulation model exactly replicated world results. Unfortunately this is not a possible. Different comparison techniques were used in this thesis to show the faithfulness of the simulation to known results. RAPTOR responded very well to all the comparisons that were applied and proved to be an efficient, reliable, and accurate simulation model.

Since RAPTOR is a generic model, it has slower running characteristics. The reason for this may be the enormous number of features. RAPTOR gives different type of results related to the system including the minimum and maximum values and standard deviation, more than that of other simulation models. The built-in animation feature is another factor for the model to run slow, however, this feature is optional and can be turned off during the runs.

Another issue for this thesis was developing a generic VV&A plan. The plan used in this verification and validation effort can be applied to most simulation models. However, the verification and the validation techniques used might differ from the ones that are used in here. There are many techniques that exist which might be applicable to specific type of simulations. The requirements or the implementation of the model will dictate the suitable techniques to be used.

5.2 Recommendations

The overall performance of RAPTOR is very pleasing. No logical, numerical, or behavioral errors were found. There were also no significant mathematical or statistical errors present. However, there are still some more evaluation and improvements to be done.

First of all, the model should be compared to real life system. It would be better if this system is an Air Force system, which is already being used, because it would be easy to get accurate data. The next thing to be checked is the code. The code of the model was not checked in this effort because it required expertise in ModSim II

simulation language. There are also more than three hundred thousand lines of code. The amount of time requirements for this job is beyond the limits of this research.

There are a few minor deficiencies in the general usability of the model. These are addressed in Chapter 4 and the recommendations to fix these deficiencies were also stated in the same chapter. These minor deficiencies have no effect on the overall system performance. The upgrades would make RAPTOR more user-friendly and efficient.

The phasing feature was tested during the evaluations. However, since it is a new add-in to RAPTOR 5.0, more testing focused directly on phasing must be accomplished to show that all sub-features under phasing are working correctly. During the comparison to a real life system, the agent should make sure that this feature is also used.

5.3 Final Thought

RAPTOR is a useful and necessary simulation tool for AFOTEC. It has been upgraded over the past few years continually to make it more effective. After the evaluations stated in the recommendations section are made, it can be accredited by the accreditation authority. RAPTOR will enhance the test and evaluation capabilities of AFOTEC.

Appendices

Appendix A. Event Log for Simple System Simulation

Starting Run 1

13.100823	a	Failed,	TimeOperated	= 13.100823	System = Yellow
43.236496	a	Repaired,	RepairTime	= 0.135673	System = Green
48.259814	c	Failed,	TimeOperated	= 48.259814	System = Red
54.730495	d	Failed,	TimeOperated	= 54.730495	System = Red
67.134390	d	Repaired,	RepairTime	= 12.403896	System = Red
81.328490	d	Failed,	TimeOperated	= 14.194100	System = Red
81.493527	c	Repaired,	RepairTime	= 33.233712	System = Red
100.000000	a	1 new spare(s) arrived			
120.135579	a	Failed,	TimeOperated	= 76.899083	System = Red
129.656798	c	Failed,	TimeOperated	= 48.163272	System = Red
153.964957	a	Repaired,	RepairTime	= 3.829377	System = Red
159.099825	c	Repaired,	RepairTime	= 29.443027	System = Red
174.396029	d	Repaired,	RepairTime	= 93.067539	System = Green
185.757239	c	Failed,	TimeOperated	= 26.657414	System = Red
195.597575	a	Failed,	TimeOperated	= 41.632619	System = Red
200.000000	a	1 new spare(s) arrived			
232.917202	a	Repaired,	RepairTime	= 7.319627	System = Red
234.734437	c	Repaired,	RepairTime	= 48.977198	System = Green
251.474548	b	Failed,	TimeOperated	= 76.361136	System = Yellow
255.197180	a	Failed,	TimeOperated	= 22.279978	System = Red
277.140191	b	Repaired,	RepairTime	= 25.665643	System = Yellow
292.279176	a	Repaired,	RepairTime	= 7.081996	System = Green
300.000000	a	1 new spare(s) arrived			
302.980969	b	Failed,	TimeOperated	= 25.840778	System = Yellow
312.896100	d	Failed,	TimeOperated	= 138.500071	System = Red
315.143100	b	Repaired,	RepairTime	= 12.162131	System = Red
327.004328	d	Repaired,	RepairTime	= 14.108229	System = Green
335.118602	c	Failed,	TimeOperated	= 100.384165	System = Red
348.067578	a	Failed,	TimeOperated	= 55.788401	System = Red
354.958730	d	Failed,	TimeOperated	= 27.954402	System = Red
363.638469	d	Repaired,	RepairTime	= 8.679739	System = Red
391.111842	a	Repaired,	RepairTime	= 13.044265	System = Red
400.000000	a	1 new spare(s) arrived			

412.500839	d	Failed,	TimeOperated	= 48.862370	System = Red
420.670363	c	Repaired,	RepairTime	= 85.551761	System = Red
429.188476	d	Repaired,	RepairTime	= 16.687636	System = Green
431.837431	d	Failed,	TimeOperated	= 2.648956	System = Red
435.399391	d	Repaired,	RepairTime	= 3.561960	System = Green
475.725559	c	Failed,	TimeOperated	= 55.055196	System = Red
500.000000	a	1 new spare(s) arrived			
516.530907	d	Failed,	TimeOperated	= 81.131516	System = Red
517.534744	a	Failed,	TimeOperated	= 126.422902	System = Red
519.084376	c	Repaired,	RepairTime	= 43.358817	System = Red
533.511974	d	Repaired,	RepairTime	= 16.981067	System = Yellow
554.653730	c	Failed,	TimeOperated	= 35.569354	System = Red
556.665747	a	Repaired,	RepairTime	= 9.131002	System = Red
569.776261	c	Repaired,	RepairTime	= 15.122531	System = Green
600.000000	a	1 new spare(s) arrived			
661.010491	d	Failed,	TimeOperated	= 127.498517	System = Red
667.370584	d	Repaired,	RepairTime	= 6.360093	System = Green
672.788508	b	Failed,	TimeOperated	= 168.883307	System = Yellow
672.788508	b	***No spare in stock***			
672.788508	b	spare ordered			
686.355527	c	Failed,	TimeOperated	= 116.579266	System = Red
692.894162	c	Repaired,	RepairTime	= 6.538635	System = Yellow
693.653197	c	Failed,	TimeOperated	= 0.759034	System = Red
696.788508	b	ordered spare arrives			
697.534107	b	Repaired,	RepairTime	= 0.745599	System = Red
700.000000	a	1 new spare(s) arrived			
712.368837	a	Failed,	TimeOperated	= 155.703091	System = Red
728.875698	d	Failed,	TimeOperated	= 61.505114	System = Red
738.888291	d	Repaired,	RepairTime	= 10.012593	System = Red
741.829283	d	Failed,	TimeOperated	= 2.940992	System = Red
750.462133	c	Repaired,	RepairTime	= 56.808937	System = Red
770.558213	c	Failed,	TimeOperated	= 20.096080	System = Red
772.302515	a	Repaired,	RepairTime	= 29.933678	System = Red
781.988140	c	Repaired,	RepairTime	= 11.429927	System = Red
792.468290	d	Repaired,	RepairTime	= 50.639007	System = Green
800.000000	a	1 new spare(s) arrived			
852.819725	a	Failed,	TimeOperated	= 80.517210	System = Yellow
886.088493	a	Repaired,	RepairTime	= 3.268768	System = Green
889.848509	c	Failed,	TimeOperated	= 107.860369	System = Red
894.107067	d	Failed,	TimeOperated	= 101.638777	System = Red
900.000000	a	1 new spare(s) arrived			

922.602775 c	Repaired,	RepairTime	= 32.754266	System = Red
937.171252 c	Failed,	TimeOperated	= 14.568477	System = Red
937.979180 a	Failed,	TimeOperated	= 51.890687	System = Red
958.960850 c	Repaired,	RepairTime	= 21.789597	System = Red
970.803923 d	Repaired,	RepairTime	= 76.696856	System = Yellow
972.801281 b	Failed,	TimeOperated	= 99.377577	System = Red

972.801281 b ***No spare in stock***

972.801281 b spare ordered

974.468993 a	Repaired,	RepairTime	= 6.489814	System = Yellow
988.946864 d	Failed,	TimeOperated	= 18.142941	System = Red
992.828837 a	Failed,	TimeOperated	= 18.359843	System = Red

996.801281 b ordered spare arrives

1000.000000 a 1 new spare(s) arrived

1000.000000 b Partial Repair, RepairTime = 3.198719
1000.000000 d Partial Repair, RepairTime = 11.053136

1000.000000 Simulation Terminated

Appendix B. Event Log for Complex System Simulation

Starting Run 1

0.949713	block5	Failed,	TimeOperated = 0.949713	System = Yellow
12.403896	block8	Failed,	TimeOperated = 12.403896	System = Yellow
13.100823	block1	Failed,	TimeOperated = 13.100823	System = Yellow
14.827253	block5	Repaired,	RepairTime = 13.877540	System = Yellow
16.221828	block12	Failed,	TimeOperated = 16.221828	System = Yellow
17.227185	block10	Failed,	TimeOperated = 17.227185	System = Yellow
20.420450	block1	Repaired,	RepairTime = 7.319627	System = Yellow
29.905199	block9	Failed,	TimeOperated = 29.905199	System = Yellow
32.156125	block5	Failed,	TimeOperated = 17.328872	System = Yellow
34.464395	block13	Failed,	TimeOperated = 34.464395	System = Red
39.093210	block4	Failed,	TimeOperated = 39.093210	System = Red
39.093210	block4	Obtained 1 Resource(s)		
39.211099	block3	Failed,	TimeOperated = 39.211099	System = Red
48.511384	block13	Repaired,	RepairTime = 14.046989	System = Yellow
49.211099	block3	Switched to a cold-standby		System = Yellow
55.793277	block12	Repaired,	RepairTime = 39.571449	System = Yellow
56.189852	block11	Failed ,	TimeOperated = 42.142863	System = Yellow
60.873718	Csp1	Incremented,	RepairTime = 11.662618	System = Yellow
62.606925	block4	Repaired,	RepairTime = 23.513714	System = Yellow
65.875710	block10	Repaired,	RepairTime = 48.648525	System = Yellow
67.876565	block2	Failed ,	TimeOperated = 67.876565	System = Yellow
69.367764	block2	Repaired,	RepairTime = 1.491199	System = Yellow
70.196822	block9	Repaired,	RepairTime = 40.291623	System = Yellow
71.288919	block5	Repaired,	RepairTime = 39.132795	System = Yellow
77.073156	block4	Failed ,	TimeOperated = 14.466231	System = Yellow
77.073156	block4	Obtained 1 Resource(s)		
84.158415	block1	Failed ,	TimeOperated = 63.737965	System = Yellow
87.741165	block12	Failed ,	TimeOperated = 31.947888	System = Yellow
93.104771	block1	Repaired,	RepairTime = 8.946357	System = Yellow
98.866256	block12	Repaired,	RepairTime = 11.125091	System = Yellow
101.893634	block12	Failed ,	TimeOperated = 3.027378	System = Yellow
104.268274	block12	Repaired,	RepairTime = 2.374640	System = Yellow
111.974849	block4	Repaired,	RepairTime = 34.901693	System = Yellow
122.523681	block1	Failed ,	TimeOperated = 29.418910	System = Yellow
128.289313	block1	Repaired,	RepairTime = 5.765632	System = Yellow
129.655649	block11	Repaired,	RepairTime = 73.465797	System = Yellow
130.690406	block13	Failed ,	TimeOperated = 82.179022	System = Red
130.748024	block9	Failed ,	TimeOperated = 60.551202	System = Red
132.730388	block13	Repaired,	RepairTime = 2.039983	System = Yellow
140.276628	block4	Failed ,	TimeOperated = 28.301779	System = Yellow

140.276628	block4	Obtained 1 Resource(s)		
141.243277	block7	Failed ,	TimeOperated = 141.243277	System = Yellow
144.900936	block3	Failed ,	TimeOperated = 95.689837	System = Yellow
154.900936	block3	Switched to a cold-standby		System = Yellow
157.703389	block2	Failed ,	TimeOperated = 88.335624	System = Yellow
168.528264	block13	Failed ,	TimeOperated = 35.797875	System = Red
170.009709	block2	Repaired,	RepairTime = 12.306320	System = Red
178.464582	block10	Failed ,	TimeOperated = 112.588871	System = Red
186.071039	block13	Repaired,	RepairTime = 17.542776	System = Yellow
186.697636	block1	Failed ,	TimeOperated = 58.408323	System = Yellow
191.999543	block1	Repaired,	RepairTime = 5.301907	System = Yellow
192.492321	block6	Failed ,	TimeOperated = 192.492321	System = Yellow
194.753415	block9	Repaired,	RepairTime = 64.005391	System = Yellow
196.990006	block12	Failed ,	TimeOperated = 92.721732	System = Yellow
197.492321	block6	Switched to a cold-standby		System = Yellow
199.633283	block3	Failed ,	TimeOperated = 44.732347	System = Yellow
201.230068	block12	Repaired,	RepairTime = 4.240062	System = Yellow
209.633283	block3	Switched to a cold-standby		System = Yellow
211.811723	block1	Failed ,	TimeOperated = 19.812180	System = Yellow
214.874106	block1	Repaired,	RepairTime = 3.062383	System = Yellow
215.407098	block7	Repaired,	RepairTime = 64.163821	System = Yellow
226.066571	block1	Failed ,	TimeOperated = 11.192466	System = Yellow
226.386296	block1	Repaired,	RepairTime = 0.319724	System = Yellow
229.561487	block8	Repaired,	RepairTime = 217.157591	System = Yellow
230.778867	Csp1	Incremented,	RepairTime = 75.877931	System = Yellow
231.346997	block4	Repaired,	RepairTime = 91.070369	System = Yellow
233.447705	block4	Failed ,	TimeOperated = 2.100708	System = Yellow
233.447705	block4	Obtained 1 Resource(s)		
236.097712	Csp1	Incremented,	RepairTime = 26.464429	System = Yellow
237.074552	block11	Failed ,	TimeOperated = 87.836144	System = Yellow
238.533383	block3	Failed ,	TimeOperated = 28.900100	System = Yellow
239.574080	block8	Failed ,	TimeOperated = 10.012593	System = Yellow
239.574080	block8	***No spare in stock***		
239.574080	sp2	spare ordered		
246.882504	block11	Repaired,	RepairTime = 9.807952	System = Yellow
247.546659	block11	Failed ,	TimeOperated = 0.664155	System = Yellow
247.546659	block11	***No spare in stock***		
247.546659	sp2	spare ordered		
248.533383	block3	Switched to a cold-standby		System = Yellow

257.325418	Csp1	Incremented, RepairTime = 8.792035	System = Yellow
271.521627	block12	Failed , TimeOperated = 70.291559	System = Yellow
274.844922	Csp1	Incremented, RepairTime = 77.352601	System = Yellow
289.528721	block6	Failed , TimeOperated = 92.036400	System = Yellow
290.569123	block10	Repaired, RepairTime = 112.104541	System = Yellow
294.528721	block6	Switched to a cold-standby	System = Yellow
300.407424	block5	Failed , TimeOperated = 209.535746	System = Yellow
300.407424	block5	***No spare in stock***	
300.407424	sp2	spare ordered	
300.561278	block12	Repaired, RepairTime = 29.039651	System = Yellow
309.062425	block1	Failed , TimeOperated = 82.676129	System = Yellow
316.307578	block1	Repaired, RepairTime = 7.245153	System = Yellow
317.846050	block4	Repaired, RepairTime = 84.398345	System = Yellow
321.108581	Csp1	Incremented, RepairTime = 26.579860	System = Yellow
330.805293	block4	Failed , TimeOperated = 12.959243	System = Yellow
330.805293	block4	Obtained 1 Resource(s)	
335.177416	block1	Failed , TimeOperated = 18.869838	System = Yellow
339.272568	block7	Failed , TimeOperated = 123.865470	System = Yellow
339.272568	block7	***No spare in stock***	
340.225688	block9	Failed , TimeOperated = 145.472273	System = Yellow
342.609858	block1	Repaired, RepairTime = 7.432442	System = Yellow
342.731891	block10	Failed , TimeOperated = 52.162768	System = Yellow
342.731891	block10	***No spare in stock***	
365.494802	block2	Failed , TimeOperated = 195.485093	System = Yellow
365.494802	block2	***No spare in stock***	
404.114417	block9	Repaired, RepairTime = 63.888729	System = Yellow
408.022775	block9	Failed , TimeOperated = 3.908359	System = Yellow
408.062387	block6	Failed , TimeOperated = 113.533665	System = Red
413.062387	block6	Switched to a cold-standby	System = Yellow
413.421094	Csp1	Incremented, RepairTime = 0.358707	System = Yellow
418.287465	block1	Failed , TimeOperated = 75.677606	System = Red
428.782128	block1	Repaired, RepairTime = 10.494664	System = Yellow
433.162427	block3	Failed , TimeOperated = 184.629044	System = Yellow
437.139872	block1	Failed , TimeOperated = 8.357744	System = Red
443.162427	block3	Switched to a cold-standby	System = Red
444.395343	block6	Failed , TimeOperated = 31.332956	System = Red
449.395343	block6	Switched to a cold-standby	System = Red

470.495704	block1	Repaired,	RepairTime = 33.355832	System = Yellow
480.309690	block3	Failed ,	TimeOperated = 37.147262	System = Yellow
488.250973	Csp1	Incremented,	RepairTime = 38.855630	System = Yellow
490.309690	block3	Switched to a cold-standby		System = Yellow
490.351339	Csp1	Incremented,	RepairTime = 47.188911	System = Yellow
496.683398	Csp1	Incremented,	RepairTime = 6.373709	System = Yellow
500.000000	sp1	1 new spare(s) arrived		
500.541502	block13	Failed ,	TimeOperated = 314.470463	System = Red
501.002229	block1	Failed ,	TimeOperated = 30.506524	System = Red
501.794259	block1	Repaired,	RepairTime = 0.792030	System = Red
505.086228	block12	Failed ,	TimeOperated = 204.524950	System = Red
505.086228	block12	***No spare in stock***		
505.708930	block13	Repaired,	RepairTime = 5.167428	System = Red
514.058059	block10	Repaired,	RepairTime = 14.058059	System = Red
520.600167	block4	Repaired,	RepairTime = 189.794874	System = Red
540.262584	block6	Failed ,	TimeOperated = 90.867241	System = Red
545.262584	block6	Switched to a cold-standby		System = Red
558.068913	Csp1	Incremented,	RepairTime = 12.806329	System = Red
567.871284	block1	Failed ,	TimeOperated = 66.077025	System = Red
589.268101	block9	Repaired,	RepairTime = 181.245326	System = Red
596.761054	block3	Failed ,	TimeOperated = 106.451364	System = Red
597.740353	block9	Failed ,	TimeOperated = 8.472252	System = Red
602.953988	block1	Repaired,	RepairTime = 35.082704	System = Red
606.761054	block3	Switched to a cold-standby		System = Red
617.207908	block4	Failed ,	TimeOperated = 96.607741	System = Red
617.207908	block4	Obtained 1 Resource(s)		
619.695718	block4	Repaired,	RepairTime = 2.487810	System = Red
647.630763	block10	Failed ,	TimeOperated = 133.572704	System = Red
647.630763	block10	***No spare in stock***		
656.684868	block4	Failed ,	TimeOperated = 36.989150	System = Red
656.684868	block4	Obtained 1 Resource(s)		
677.577245	block3	Failed ,	TimeOperated = 70.816191	System = Red
679.292885	block13	Failed ,	TimeOperated = 173.583955	System = Red
681.821194	block13	Repaired,	RepairTime = 2.528308	System = Red
687.577245	block3	Switched to a cold-standby		System = Red
702.026660	block1	Failed ,	TimeOperated = 99.072672	System = Red
716.309152	block3	Failed ,	TimeOperated = 28.731907	System = Red
724.836217	block9	Repaired,	RepairTime = 127.095864	System = Red

726.309152	block3	Switched to a cold-standby		System = Red
730.210666	block9	Failed ,	TimeOperated = 5.374448	System = Red
731.863644	block1	Repaired,	RepairTime = 29.836984	System = Red
733.088233	block3	Failed ,	TimeOperated = 6.779081	System = Red
734.579113	block6	Failed ,	TimeOperated = 189.316529	System = Red
739.579113	block6	Switched to a cold-standby		System = Red
742.581216	Csp1	Incremented,	RepairTime = 16.272063	System = Red
743.088233	block3	Switched to a cold-standby		System = Red
746.824378	Csp1	Incremented,	RepairTime = 3.736145	System = Red
752.460606	block1	Failed ,	TimeOperated = 20.596962	System = Red
754.157292	block1	Repaired,	RepairTime = 1.696685	System = Red
761.750003	block4	Repaired,	RepairTime = 105.065135	System = Red
762.204427	block4	Failed ,	TimeOperated = 0.454425	System = Red
762.204427	block4	Obtained 1 Resource(s)		
763.520971	block6	Failed ,	TimeOperated = 23.941858	System = Red
767.107049	Csp1	Incremented,	RepairTime = 79.529804	System = Red
768.520971	block6	Switched to a cold-standby		System = Red
768.633012	Csp1	Incremented,	RepairTime = 0.112041	System = Red
771.730830	block1	Failed ,	TimeOperated = 17.573539	System = Red
779.622018	block1	Repaired,	RepairTime = 7.891187	System = Red
783.151028	block3	Failed ,	TimeOperated = 40.062795	System = Red
793.151028	block3	Switched to a cold-standby		System = Red
794.960127	Csp1	Incremented,	RepairTime = 55.381014	System = Red
796.654737	block1	Failed ,	TimeOperated = 17.032720	System = Red
796.865425	block1	Repaired,	RepairTime = 0.210688	System = Red
812.788858	Csp1	Incremented,	RepairTime = 206.027804	System = Red
819.089579	block4	Repaired,	RepairTime = 56.885151	System = Red
835.198979	block13	Failed ,	TimeOperated = 153.377785	System = Red
836.538965	block13	Repaired,	RepairTime = 1.339986	System = Red
848.489064	block9	Repaired,	RepairTime = 118.278398	System = Red
854.745515	block9	Failed ,	TimeOperated = 6.256451	System = Red
865.632119	block6	Failed ,	TimeOperated = 97.111148	System = Red
867.132838	block4	Failed ,	TimeOperated = 48.043259	System = Red
867.132838	block4	Obtained 1 Resource(s)		
870.632119	block6	Switched to a cold-standby		System = Red
880.912656	block4	Repaired,	RepairTime = 13.779818	System = Red
884.688694	block3	Failed ,	TimeOperated = 91.537666	System = Red
891.881636	block4	Failed ,	TimeOperated = 10.968980	System = Red
891.881636	block4	Obtained 1 Resource(s)		

893.717018 Csp1	Incremented, RepairTime = 23.084899	System = Red
894.688694 block3	Switched to a cold-standby	System = Red
895.261121 Csp1	Incremented, RepairTime = 0.572427	System = Red
900.869904 block6	Failed , TimeOperated = 30.237785	System = Red
905.869904 block6	Switched to a cold-standby	System = Red
906.180017 block6	Failed , TimeOperated = 0.310113	System = Red
909.680318 block13	Failed , TimeOperated = 73.141352	System = Red
911.180017 block6	Switched to a cold-standby	System = Red
913.235788 Csp1	Incremented, RepairTime = 2.055771	System = Red
914.451663 block13	Repaired, RepairTime = 4.771345	System = Red
931.888298 Csp1	Incremented, RepairTime = 138.737270	System = Red
943.150477 Csp1	Incremented, RepairTime = 37.280573	System = Red
957.716703 block1	Failed , TimeOperated = 160.851278	System = Red
968.046458 block6	Failed , TimeOperated = 56.866441	System = Red
969.873181 block13	Failed , TimeOperated = 55.421518	System = Red
970.803208 block1	Repaired, RepairTime = 13.086505	System = Red
971.887454 block13	Repaired, RepairTime = 2.014272	System = Red
973.046458 block6	Switched to a cold-standby	System = Red
977.219607 block4	Repaired, RepairTime = 85.337971	System = Red
1000.000000 sp1	1 new spare(s) arrived	
1000.000000 block6	Partial Repair, RepairTime = 26.953542	
1000.000000 block6	Cold spare partial repair cost not calculated	
1000.000000 block7	Partial Repair, RepairTime = 0.000000	
1000.000000 block9	Partial Repair, RepairTime = 145.254485	

1000.000000 Simulation Terminated

Appendix C. Event Log for Phasing Simulation

Starting Run 1

0.407020	block5	Failed ,	TimeOperated = 0.407020	System = Yellow
7.278235	block1	Failed ,	TimeOperated = 7.278235	System = Yellow
24.925284	block7	Failed ,	TimeOperated = 24.925284	System = Red
26.425121	block1	Repaired,	RepairTime = 19.146886	System = Red
27.365247	block4	Failed ,	TimeOperated = 27.365247	System = Red
30.162384	block3	Failed ,	TimeOperated = 30.162384	System = Red
33.955898	block7	Repaired,	RepairTime = 9.030613	System = Yellow
48.038406	block4	Repaired,	RepairTime = 20.673159	System = Yellow
55.135456	block4	Failed ,	TimeOperated = 7.097050	System = Yellow
60.217418	block5	Repaired,	RepairTime = 59.810398	System = Yellow
66.966167	block3	Repaired,	RepairTime = 36.803783	System = Yellow
67.876565	block2	Failed ,	TimeOperated = 67.876565	System = Yellow
83.938548	block7	Failed ,	TimeOperated = 49.982651	System = Red
85.103750	block2	Repaired,	RepairTime = 17.227185	System = Red
100.000000	System	Changing to Phase 2		
102.760579	block7	Repaired,	RepairTime =18.822031	System = Yellow
126.094064	block5	Failed ,	TimeOperated =65.876646	System = Yellow
128.328214	block6	Failed ,	TimeOperated =128.328214	System = Yellow
142.071284	block1	Failed ,	TimeOperated =115.646163	System = Red
143.422936	block5	Repaired,	RepairTime =17.328872	System = Red
156.475432	block6	Repaired,	RepairTime =28.147218	System = Yellow
175.115419	block6	Failed ,	TimeOperated =18.639987	System = Red
177.481264	block1	Repaired,	RepairTime =35.409980	System = Yellow
191.678349	block6	Repaired,	RepairTime =16.562930	System = Yellow
200.000000	sparespool	1 new spare(s) arrived		
200.000000	System	Changing to Phase 3		
210.248021	block4	Repaired,	RepairTime =155.112565	System = Green
279.498057	block4	Failed ,	TimeOperated =69.250035	System = Yellow
300.000000	System	Changing to Phase 4		
303.011771	block4	Repaired,	RepairTime =23.513714	System = Green
310.788139	block5	Failed ,	TimeOperated =167.365203	System = Yellow
316.988972	block4	Failed ,	TimeOperated =13.977201	System = Yellow
331.455203	block4	Repaired,	RepairTime =14.466231	System = Yellow
355.886388	block4	Failed ,	TimeOperated =24.431185	System = Yellow
383.699116	block4	Repaired,	RepairTime =27.812727	System = Yellow
385.023593	block4	Failed ,	TimeOperated =1.324478	System = Yellow
390.960194	block4	Repaired,	RepairTime =5.936601	System = Yellow

400.000000	sparespool	1 new spare(s) arrived	
400.000000	System	Changing to Phase 5	
409.117063	block5	Repaired, RepairTime =98.328924	System = Green
431.525952	block4	Failed , TimeOperated =40.565758	System = Yellow
459.827731	block4	Repaired, RepairTime =28.301779	System = Green
500.000000	System	Changing to Phase 6	
523.576989	block4	Failed , TimeOperated =63.749258	System = Yellow
534.177144	block4	Repaired, RepairTime =10.600155	System = Green
564.929701	block4	Failed , TimeOperated =30.752557	System = Yellow
581.617356	block4	Repaired, RepairTime =16.687655	System = Green
583.087852	block4	Failed , TimeOperated =1.470496	System = Yellow
583.087852	block4	***No spare in stock***	
600.000000	sparespool	1 new spare(s) arrived	
600.000000	System	Changing to Phase 7	
684.398345	block4	Repaired, RepairTime =84.398345	System = Green
700.000000	System	Changing to Phase 8	
735.217733	block4	Failed , TimeOperated =50.819388	System = Yellow
735.217733	block4	***No spare in stock***	
800.000000	sparespool	1 new spare(s) arrived	
800.000000	System	Changing to Phase 9	
900.000000	System	Changing to Phase 10	
927.828094	block4	Repaired, RepairTime =127.828094	System = Green
936.899564	block4	Failed , TimeOperated =9.071470	System = Yellow
936.899564	block4	***No spare in stock***	
960.823375	block6	Failed , TimeOperated =769.145026	System = Yellow
997.476830	block6	Repaired, RepairTime =36.653455	System = Yellow
1000.000000	sparespool	1 new spare(s) arrived	
1000.000000	block4	Partial Repair, RepairTime=0.000000	

1000.000000 Simulation Terminated

Appendix D. End of Run Report for Comp3 Simulation

End of Run #1

Ending SimTime= 1000.000000
Number of system failures = 6
Number of component failures = 64

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	**162.096243**	28.309332	306.189571	98.942155
Down Times	4.570424	1.067708	8.953001	2.695389
Total Time Between Component Failures	15.537531	0.344740	54.381696	11.615070
Component Repair Times	11.998220	3.689210	29.079950	6.418175

Availability = **0.972577457**
MTBM = **15.196523**
GreenTime = 40.587278%
YellowTime = 56.670468%
RedTime = **2.742254%**

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	916.55	83.45	8.00	1009.00
2	1.00	798.30	201.70	10.00	1011.00
5	1.00	910.12	89.88	12.00	1013.00
3	1.00	926.09	73.91	15.00	1016.00
4	1.00	820.55	179.45	12.00	1013.00
6	1.00	945.32	54.68	3.00	1004.00
7	1.00	921.58	78.42	4.00	1005.00
Block Totals	7.00	6238.51	761.49	64.00	7071.00

Grand Total 7071.00

****

End of Run #2

Ending SimTime= 1000.000000
Number of system failures = 9
Number of component failures = 74

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	103.860910	21.068148	314.622511	88.340325
Down Times	7.250201	1.485426	12.115139	3.569893
Total Time Between Component Failures	12.911986	0.398413	38.128766	9.613922
Component Repair Times	12.012773	3.995005	25.908401	6.183710

Availability = 0.934748188
MTBM = 12.631732
GreenTime = 33.962320%
YellowTime = 59.512499%
RedTime = 6.525181%

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	906.33	93.67	8.00	1009.00
2	1.00	815.11	184.89	9.00	1010.00
5	1.00	860.23	139.77	17.00	1018.00
3	1.00	902.72	97.28	18.00	1019.00
4	1.00	821.41	178.59	13.00	1014.00
6	1.00	855.33	144.67	7.00	1008.00
7	1.00	949.91	50.09	2.00	1003.00
Block Totals	7.00	6111.05	888.95	74.00	7081.00

Grand Total 7081.00

****

End of Run #3

Ending SimTime= 1000.000000
Number of system failures = 7
Number of component failures = 76

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	138.433886	17.609651	161.085223	50.304496
Down Times	**4.423257**	0.318296	8.282756	2.631559
Total Time Between Component Failures	12.977218	0.091552	58.077537	11.896779
Component Repair Times	11.747400	3.163824	29.853451	7.465307

Availability = 0.969037201
MTBM = 12.750489
GreenTime = 36.683760%
YellowTime = 60.219960%
RedTime = 3.096280%

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	883.05	116.95	11.00	1012.00
2	1.00	889.31	110.69	6.00	1007.00
5	1.00	906.21	93.79	13.00	1014.00
3	1.00	898.65	101.35	22.00	1023.00
4	1.00	866.17	133.83	10.00	1011.00
6	1.00	878.26	121.74	6.00	1007.00
7	1.00	785.54	214.46	8.00	1009.00
Block Totals	7.00	6107.20	892.80	76.00	7083.00

Grand Total 7083.00

****

End of Run #4

Ending SimTime= 1000.000000
Number of system failures = 10
Number of component failures = 72

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	94.315696	0.956191	175.971563	63.936751
Down Times	5.684304	0.918555	10.768905	2.829054
Total Time Between Component Failures	13.751471	0.069562	56.287808	12.268698
Component Repair Times	**13.786441**	3.719212	28.769461	6.496083

Availability = 0.943156964
MTBM = 13.099402
GreenTime = 32.493553%
YellowTime = 61.822143%
RedTime = 5.684304%

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	842.90	157.10	16.00	1017.00
2	1.00	883.31	116.69	6.00	1007.00
5	1.00	920.59	79.41	10.00	1011.00
3	1.00	955.53	44.47	9.00	1010.00
4	1.00	776.73	223.27	14.00	1015.00
6	1.00	837.99	162.01	8.00	1009.00
7	1.00	790.33	209.67	9.00	1010.00
Block Totals	7.00	6007.38	992.62	72.00	7079.00

Grand Total 7079.00

****

End of Run #5

Ending SimTime= 1000.000000
Number of system failures = 8
Number of component failures = 76

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	119.803204	1.341078	266.142503	103.758315
Down Times	5.196796	1.422675	11.275502	3.071265
Total Time Between Component Failures	12.899445	0.211473	72.523181	13.322223
Component Repair Times	11.760887	3.404244	23.579041	5.831966

Availability = 0.958425631
MTBM = 12.610864
GreenTime = 35.291434%
YellowTime = 60.551130%
RedTime = 4.157437%

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	855.49	144.51	15.00	1016.00
2	1.00	858.14	141.86	7.00	1008.00
5	1.00	889.39	110.61	14.00	1015.00
3	1.00	931.00	69.00	14.00	1015.00
4	1.00	790.08	209.92	15.00	1016.00
6	1.00	827.20	172.80	9.00	1010.00
7	1.00	954.87	45.13	2.00	1003.00
Block Totals	7.00	6106.17	893.83	76.00	7083.00

Grand Total 7083.00

****

End of Run #6

Ending SimTime= 1000.000000
Number of system failures = 8
Number of component failures = 68

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	115.857973	5.721780	550.820778	170.616769
Down Times	**9.142027**	0.260558	18.133364	5.990827
Total Time Between Component Failures	14.705756	0.101464	61.234672	13.777678
Component Repair Times	13.413348	3.623280	29.521524	6.754815

Availability = **0.926863784**
MTBM = 13.630350
GreenTime = 41.623129%
YellowTime = 51.063249%
RedTime = **7.313622%**

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	888.33	111.67	11.00	1012.00
2	1.00	802.25	197.75	11.00	1012.00
5	1.00	913.54	86.46	12.00	1013.00
3	1.00	947.51	52.49	11.00	1012.00
4	1.00	809.15	190.85	12.00	1013.00
6	1.00	876.66	123.34	6.00	1007.00
7	1.00	874.73	125.27	5.00	1006.00
Block Totals	7.00	6112.18	887.82	68.00	7075.00

Grand Total 7075.00

****

End of Run #7

Ending SimTime= 1000.000000
Number of system failures = 12
Number of component failures = 79

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	**78.320361**	1.372250	237.603831	71.580889
Down Times	5.012972	0.677053	17.199420	4.324360
Total Time Between Component Failures	12.629687	0.000547	109.502728	5.499127
Component Repair Times	**11.244902**	2.693152	35.713809	6.843903

Availability = 0.939844337
MTBM = **11.896764**
GreenTime = **44.738310%**
YellowTime = **49.246124%**
RedTime = 6.015566%

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	888.67	111.33	12.00	1013.00
2	1.00	806.13	193.87	10.00	1011.00
5	1.00	837.37	162.63	20.00	1021.00
3	1.00	898.35	101.65	21.00	1022.00
4	1.00	924.10	75.90	5.00	1006.00
6	1.00	834.82	165.18	8.00	1009.00
7	1.00	921.15	78.85	3.00	1004.00
Block Totals	7.00	6110.59	889.41	79.00	7086.00

Grand Total 7086.00

****

End of Run #8

Ending SimTime= 1000.000000
Number of system failures = 8
Number of component failures = 73

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	119.833343	20.269835	373.362701	109.031475
Down Times	5.166657	1.284684	14.271899	3.901198
Total Time Between Component Failures	13.605911	0.377931	51.023985	10.194772
Component Repair Times	11.604847	3.681945	26.672368	6.292389

Availability = 0.958666740
MTBM = 13.132421
GreenTime = 38.589838%
YellowTime = 57.276836%
RedTime = 4.133326%

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	908.08	91.92	9.00	1010.00
2	1.00	824.07	175.93	9.00	1010.00
5	1.00	850.42	149.58	19.00	1020.00
3	1.00	906.69	93.31	18.00	1019.00
4	1.00	874.50	125.50	8.00	1009.00
6	1.00	888.59	111.41	6.00	1007.00
7	1.00	905.33	94.67	4.00	1005.00
Block Totals	7.00	6157.68	842.32	73.00	7080.00

Grand Total 7080.00

****

End of Run #9

Ending SimTime= 1000.000000
Number of system failures = 9
Number of component failures = 69

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	106.345797	3.090816	187.997467	70.402540
Down Times	4.765315	0.524584	15.584759	4.469667
Total Time Between Component Failures	14.335319	0.131159	45.606254	10.461154
Component Repair Times	13.644101	4.101679	34.032733	7.110441

Availability = 0.957112168
MTBM = 13.871191
GreenTime = **31.946697%**
YellowTime = **63.764520%**
RedTime = 4.288783%

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	863.76	136.24	12.00	1013.00
2	1.00	900.59	99.41	5.00	1006.00
5	1.00	836.19	163.81	19.00	1020.00
3	1.00	955.47	44.53	9.00	1010.00
4	1.00	871.49	128.51	8.00	1009.00
6	1.00	862.91	137.09	7.00	1008.00
7	1.00	768.16	231.84	9.00	1010.00
Block Totals	7.00	6058.56	941.44	69.00	7076.00

Grand Total 7076.00

****

End of Run #10

Ending SimTime= 1000.000000
Number of system failures = 9
Number of component failures = 77

	Mean	Minimum	Maximum	St.Dev
Time Between Downing Events	104.503123	0.401614	254.487800	84.983994
Down Times	6.607988	1.373503	19.198188	5.164669
Total Time Between Component Failures	12.932849	0.089354	46.210190	11.484511
Component Repair Times	12.742949	3.687013	34.604066	6.830859

Availability = 0.940528110
MTBM = 12.214651
GreenTime = 34.773866%
YellowTime = 59.278945%
RedTime = 5.947189%

Cost Data

Block	InitCost	OpCost	RepCost	IndvSpCost	Total
1	1.00	885.83	114.17	11.00	1012.00
2	1.00	731.16	268.84	14.00	1015.00
5	1.00	886.37	113.63	15.00	1016.00
3	1.00	919.71	80.29	16.00	1017.00
4	1.00	858.83	141.17	9.00	1010.00
6	1.00	809.69	190.31	10.00	1011.00
7	1.00	943.20	56.80	2.00	1003.00
Block Totals	7.00	6034.78	965.22	77.00	7084.00

Grand Total 7084.00

****

Bibliography

1. Locks, M. Reliability, Maintainability, and Availability Assessment. Rochelle Park NJ: Hayden Book Company, 1983.

2. Kececioglu, D. Reliability and Life Testing Handbook. Englewood Cliffs NJ: Prentice Hall, 1993.

3. Hausen, H. L. Software Validation. New York: Elsevier Science Publishing Company, 1983.

4. Smith, D. Reliability and Maintainability in Perspective. New York: John Wiley and Sons, 1985.

5. Andriole, S. Software Validation Verification Testing and Documentation. Princeton NJ: Petrocelli Books, 1982.

6. Miller,E. and Howden, W. Software Testing and Validation Techniques. New York: Computer Society Press, 1982.

7. Department of Defense. Dod Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A). DoD Instruction 5000.61. Washington: GPO, 29 April 1996.

8. " User's Manual." Rapid Availability Prototyping for Operational Readiness. Version 2.0. New Mexico, 16 Oct 1995.

9. "RAPTOR Specifications." Excerpt from unpublished article. n. pag. <http://www.arinc.com/raptor>. 20 July 2000.

10. Department of the Air Force. Verification, Validation and Accreditation (VV&A). AFI 16- 1001. Washington: HQ USAF, 1 June 1996.

11. Youngblood, S. M. and Pace, K. D. "An Overview of Model and Simulation Verification, Validation, and Accreditation," John Hopkins APL Technical Digest, Volume 16, Number 2: 197-205 (1995).

12. Ketanni, O. and Oral, M. "The Facets of the Modeling and Validation Process in Operations Research," European Journal of Operational Research, Volume 66, Number 2: 216-234 (16 April1993).

13. Law, M. A. and Kelton, D. V. Simulation Modeling and Analysis. New York: McGraw Hill, 1982.

14. Balci, O. "Verification, Validation and Accreditation of Simulation Models," Proceedings of Winter Simulation Conference. 135-141. 1997.

15. Sargent, G. R. "Verification and Validation of Simulation Models," Proceedings of Winter Simulation Conference. 121-130. 1998.

16. Yourdon, E. Managing the Structured Techniques. Englewood Cliffs NJ: Prentice-Hall, 1979.

17. Deutsch, S. M. Software Verification and Validation. Englewood Cliffs NJ: Prentice-Hall, 1982.

18. "Verification, Validation, and Accreditation." Excerpt from unpublished article. n. pag. < http://www.msiac.dmso.mil/vva>. 02 Nov 2000.

19. Kleijnen, J. P. C. "Verification and Validation of Simulation Models," European Journal of Operational Research, Volume 82: 145-162 (1995).

20. "Validation." Excerpt from unpublished article. n. pag. <http://hissa.nist.govt/HHRFdata/Artifacts/ITLdoc/234/val-proc.html>. 10 Nov 2000.

21. Whitner, R. B. and Balci O. "Guidelines for Selecting and Using Simulation Verification Techniques," Proceedings of Winter Simulation Conference. 559-568. Washington D.C., 1989.

22. Department of Defense. Modeling and Simulation (M&S) Master Plan. DoD Directive 5000.59. Washington: GPO, October 1995.

23. Balci, O. and Sargent R. G. "Validation of Simulation Models via Simultaneous Confidence Intervals," American Journal of Mathematical and Management Science, Volume 4, Number 3: 375-406 (1984).

24. Kleijnen J. Statistical Tools for Simulation Practitioners. New York: Marcel Dekker, 1987.

25. Ross M. S. Introduction to Probability Models. Chestnut Hill ME: Academic Press, 1997.

26. Robinson S. "Simulation Model Verification and Validation: Increasing the User's Confidence," Proceedings of Winter Simulation Conference. 53-59. 1997.

27. Wackerly, D. and Mendenhall, W. and Scheaffer, R. Mathematical Statistics with Applications. California: Duxbury Press, 1996.

28. Sargent G. R. " A Tutorial on Validation and Verification of Simulation Models," Proceedings of Winter Simulation Conference. 33-39. 1988.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
02-26-01	Master's Thesis	Aug 2000 – Mar 2001

4. TITLE AND SUBTITLE

A VERIFICATION AND VALIDAITON ASSESSMENT OF RAPID AVAILABILITY PROTOTYPING FOR TESTING OPERATIONAL READINESS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Tarman, Gurhan, 1st Lieutenant, TUAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GOR/ENS/01M-15

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Through out the world, new techniques are developed to test and evaluate systems. The Department of Defense and the US Air Force conduct numerous studies into the reliability and maintainability of current and future weapons systems in an effort to understand the reliability, maintainability, and availability (RM&A) costs of these systems. Furthermore, they must verify RM&A characteristics of systems, which are still in the development phase. The Air Force Operational Test and Evaluation Center (AFOTEC) is responsible for testing new equipment to ensure their RM&A characteristics meet specifications.
Rapid Availability Prototyping for Testing Operational Readiness (RAPTOR) is the software tool developed by HQ AFOTEC to enhance existing analysis capabilities. RAPTOR is a modeling framework allowing quick creation of RM&A models for many systems.
DoD has made verification, validation, and accreditation (VV&A) of these simulation models an official requirement. This research outlines a generic VV&A plan that uses the appropriate tools and techniques to perform model verification, validation, and accreditation. This plan is then applied to RAPTOR 5.0 and a portion of verification and validation of RAPTOR is completed and recommendations for accreditation are made.

15. SUBJECT TERMS

Modeling and Simulation, Reliability, Maintainability, and Availability, Verification, Validation, and Accreditation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Chambal, Stephen P., Capt.
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	128	255-6565 Ext. 4314 Email: Stephen.Chambal@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18